AFRL-IF-RS-TR-2001-225
Final Technical Report
October 2001

INTENSIVE KNOWLEDGE DISCOVERY FROM HETEROGENEOUS DISTRIBUTED DATA AND KNOWLEDGE

Capraro Technologies, Inc.

Gerard T. Capraro and Gerald B. Berdan

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2001-225 has been reviewed and is approved for publication.

APPROVED:

JOHN SPINA
Project Engineer

FOR THE DIRECTOR:

MICHAEL L. TALBERT, Technical Advisor
Information Technology Division
Information Directorate

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
OCTOBER 2001

3. REPORT TYPE AND DATES COVERED
Final Aug 00 - Jul 01

4. TITLE AND SUBTITLE
INTENSIVE KNOWLEDGE DISCOVERY FROM HETEROGENEOUS DISTRIBUTED DATA AND KNOWLEDGE

5. FUNDING NUMBERS
C - F30602-99-C-0266
PE - 61102F
PR - 5581
TA - 27
WU - 15

6. AUTHOR(S)
Gerard T. Capraro and Gerald B. Berdan

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Capraro Technologies, Inc.
311 Turner Street Suite 410
Utica New York 13501-1731

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory/IFTD
525 Brooks Road
Rome New York 13441-4514

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2001-225

11. SUPPLEMENTARY NOTES
Air Force Research Laboratory Project Engineer: John Spin/IFTD/(315) 330-4032

12a. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
An infrastructure to extend the Joint Battlespace Infosphere. (JBI) Publish and Subscribe paradigm is described. The design emphasizes the client side of the JBI architecture where the users' computing device can vary substantially in bandwidth, processing capability, screen size, and connection type. Today's Internet user can connect with the net at any time using a wide variety of devices. Our emphasis is on designing a JBI extended infrastructure that will intelligently support the efficient rapid access and dissemination of distributed knowledge base repositories to a client independent of computing device or connection. Hand-held computing devices are depicted as innovative devices for interacting with massive amounts of data and knowledge. Our goal is to design and develop an infrastructure that will efficiently and rapidly provide data, knowledge and information to a dynamic user. A hypothetical JBI architecture is described with an email publish and subscribe paradigm. A Air Mobility Command scenario is described involving 3-D images, digital terrain files, and virtual reality rendering software. A computer architecture is presented along with sample code illustrating how World Wide Web Consortium (W3C) technologies (e.g. XML, XSL) can be used to meet JBI's needs.

14. SUBJECT TERMS
Joint Battlespace Infosphere, JBI, Publish, Subscribe, Virtual Reality, World Wide Web Consortium, W3C, Knowledge Bases, Internet, Palm Pilot

15. NUMBER OF PAGES
52

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS/TABLES

Section 1 Introduction

The USAF Research Laboratory has been investigating the integration of large databases and knowledge bases for more than ten years (1-13). However, if we are to meet the goals of Vision 2010, we need the government to solve problems that cannot or will not be solved by commercial products alone. We plan on showing, through building a demonstration, that a portion of the Expeditionary Force Experiment's (EFX) requirements are obtainable and demonstrate what research is needed to meet some of their needs in 2010.

In order to meet EFX's requirements, military personnel must be ready to mobilize on a moment's notice, to communicate with superiors throughout their mission, and be kept informed continuously as events and scenarios change. They must be able to reach back and gather information while they are performing their missions and they need to provide data and information back to their command centers and to other forces in the area. Over the years these communications have been primarily audio. However, with the advances in computing and communications it is now possible to communicate to anyone with a laptop computer, a consumer electronics (CE) device, or a personal digital assistant (PDA), given a phone line or a radio frequency (RF) modem (e.g. cell phone, pagers, satellite phone). One can stay in touch with his/her e-mail, send or receive faxes, access applications on a home computer, and query knowledge and databases anywhere in the world. He/she can have access to very large amounts of data and information in any form (i.e. voice, graphics, and video). It was previously shown feasible by Capraro Technologies, Inc. (CTI) that multiple databases could be accessed over the web and presented to a hand-held computing device (HCD) using hard wired and cellular phone connections. We believe that we can extend this capability and create a demonstration for meeting some of EFX's requirements. Those requirements that cannot be met using today's technology will be highlighted and solutions proposed.

Vision 2010 states that "Information technology will improve the ability to see, prioritize, assign, and assess information." To accomplish this, we must be able to fuse all sources of intelligence. These sources may be, for example, from remote sensors, air and space platforms, command organizations, and logistic support centers. Fusing these sources intelligently using information technology will allow a greater number of operational tasks to be accomplished faster than we can today. Vision 2010 claims that "real-time information will likely drive parallel, not sequential, planning and real-time, not prearranged, decision-making. We must have information superiority: the capabilities to collect, process, and disseminate an uninterrupted flow of information while exploiting or denying an adversary's ability to do the same."

To meet the demands of information technology stated in Vision 2010, we must push the state-of-the-art on many fronts: i.e., communications, sensors, processors, memory, computing, and algorithms, to name just a few. Programs like Sensor to Shooter, EFX, Defensive Information Warfare, and Information Countermeasures For Biological Warfare require information, real-time knowledge discovery, and learning processes. These programs and their requirements will change over time based upon different

scenarios and enemies. How we design and build an information infrastructure that can meet the varied individual system user needs and requirements of accuracy, speed, precision, size, weight, power, processing, bandwidth, etc. is not known. We need the tools and the know-how for designing and evaluating their potential.

This final report provides a description of our findings and an overview of our demonstrations. The three original tasks are:

Task 1. Perform a study of one challenge problem (e.g. Sensor to Shooter, EFX) and their data, knowledge, and information needs will be quantified.

Task 2. Build a demonstration including sensor visualization, terrain modeling, message traffic, large databases, and varied size computing devices. This demonstration will leverage state-of-the-art commercial computer software tools such as Java.

Task 3. Analyze the results of the demonstration and propose new architecture designs and information processing models for organizing, searching, indexing, and fusing data, information, and knowledge.

During the course of the effort, a fourth task was added to evaluate the Joint Battlespace Infosphere (JBI) models, study the architectural designs and models of task 3, and determine their role in meeting JBI requirements.

Section 2 of this report provides a discussion of what we have done in accomplishing Task 1. Section 3 provides what we have done in accomplishing Task 2. Section 4 reports on our efforts and results of Task 3. Section 5 discusses our results of Task 4. The last section will provide a summary of the report and our suggestions for future work.

Section 2 Challenge Problem

Choosing a Challenge Problem

Capraro Technologies, Inc. (CTI) is very much aware of the work that the Information Directorate is pursuing related to the EFX and the Information For Global Reach (IFGR) contract (F30602-C-0252). CTI is a subcontractor to Litton TASC on the IFGR contract and is assisting them in the development of the Information Manager portion of the architecture. We are designing and building the software for handling messages from the communications hardware and the user front-end. We have participated in, and are aware of the JEFX-99 Scenario and were a team member on JEFX-2000.

Given the length of this effort, we chose to leverage our current knowledge with JEFX. We are using the current JEFX-99 scenario as the basis for understanding this challenge problem. We have expanded and projected the USAF's requirements based upon the information we have gathered through Litton TASC and our IFGR contract. We have projected JEFX-99 requirements based upon commercial and Government technology

enhancements in multi-media databases, data streaming, video, and Internet technology. The following is a brief overview of the Air Mobility Command issues we are addressing.

Air Mobility Command Issues

The IFGR effort is a project to develop a capability, a hardware and software system, which supports and enhances the flow of information over disadvantaged wireless communication links. In the IFGR development process, we became involved in demonstrating our technology to Air Mobility Command (AMC). We sent and received digital messages to and from their aircraft. We demonstrated a capability for them to send digital messages indicating when an aircraft has departed an airport, scheduled position reports during flight, and where and when it has landed. In addition we have sent and received weather maps, text messages requesting diversions to other locations, messages requesting non-standard text, and other command and control messages. A purpose of IFGR is to arbitrate between multiple users' data transfer needs and the available wireless communication systems, choose the proper media, and ensure that the messages reach their destination.

This effort is currently in process and CTI has used samples of these messages for other demonstrations we have built on a separate USAF effort (14). At meetings with AMC personnel it became evident that one of the capabilities that AMC would like would be the ability to send up-to-date information about an airport directly to the pilot in flight. They currently have flight plates that are carried with them. These flight plates are maps obtained from flight information publications (FLIPs) that are created about airports and are updated periodically by the National Imagery and Mapping Agency (NIMA). They contain airport diagrams, instrument approach procedures, military departure procedures, and radar instrument approach minimums. Pilots are sometimes gone for extended periods of time and may not have the most current flight plates, or they may be diverted to an airport for which they do not have the required flight plate. During the Kosovo campaign, USAF was faxing copies of flight plates to pilots on an as-needed basis. One of the tasks we are working on in the IFGR effort is to scan the flight plates, compress the image, and then send their resultant digital files to the pilots either in the cockpit or at a current ground location.

One of the major problems that we have in communicating with the aircraft is lack of bandwidth. Limited bandwidth will always be an issue with military communications to aircraft, space, or any vehicle that has to rely on Radio Frequencies (RF) for part of its communications link. However, with the advent of faster computers (doubling clock speeds every eighteen months), new compression techniques, data streaming, virtual machines, portable software, and other computing enhancements, there are new ways to minimize the impact of the bandwidth constraints.

It was while studying the flight plate problem that we envisioned a capability that would benefit AMC, along with other military organizations concerned with sharing terrain related data (e.g. battle damage assessment, flight simulation, flight training, and war

gaming). Our defined goal is to leverage commercial products and build a capability to create virtual reality simulations of flying an aircraft into an airport, given the latest data on the condition of that airport. In so doing, we will be able to assess the hardware and software required. We will be able to evaluate the process of creating the simulations on the fly, investigate and determine the best way to provide the information over finite bandwidths, and direct where enhancements in technology are needed to meet our goal.

Our Vision for AMC

Our 2010 vision for AMC would include a capability to create a simulated flight path for any airport or landing strip in the world. We can create this simulation using terrain data from either the NIMA and/or the US Geological Survey's (USGS) databases along with recent photographs from space or aircraft sensors. An individual would have the capability to warp these photographs upon the 3-D renderings of the airstrips created from the databases. The person would be able to retrieve the data from databases stored at their location and/or obtain data from networked computer sources, in real time to create the required images. They would be able to acquire the flight plates for the area and have an experienced pilot perform virtual reality landings into the simulated landing strip. Different landing approaches can be tested and evaluated depending on expected weather and battle conditions. Once an approved approach(es) is decided, the information can be sent to the pilot. The information could be sent for example, as one picture, or a series of pictures, or as virtual reality modeling language (VRML) software, depending upon the bandwidth and time allotted to get the information to the pilot.

During the second phase of this effort it was brought to our attention that there existed a United States Air Force Scientific Advisory Board (USAFSAB) "Report on Information Management to Support the Warrior". We reviewed the available material (15) along with subsequent documents related to the Joint Battlespace Infosphere (16) and concluded that there are major technological intersections and similar goals between our efforts and what they have proposed in their reports. We are only demonstrating a portion of the JBI design and will highlight those portions by using JBI terminology where appropriate in the following sections.

Joint Battlespace Infosphere

The Joint Battlespace Infosphere (JBI) is a Department of Defense (DOD) information management system (15,16). The following is a brief overview of the system obtained from the references. The result of our effort addresses some of the core issues of a JBI providing information to a user.

The JBI integrates and assembles data from multiple sources and distributes resultant information in the proper form to the appropriate level of personnel. The JBI is built upon a collection of protocols, processes, and core functions that allow for the sharing and exchanging of information. The JBI attempts to integrate legacy "stovepipe" information systems by acting as an intermediary between them so that they can share consistent information. In addition by acting as an intermediary between systems it

attempts to enhance the general pool of information through synergistic efforts between the individual systems' information. It also filters and presents information relative to the individual user's profile and needs.

The JBI architecture is based broadly upon four key concepts along with numerous supporting technologies. The four concepts are:

1. Information exchange via a publish and subscribe paradigm,
2. Data are transformed into knowledge via fuselets,
3. Collaboration between distributed clients is via updateable knowledge objects, and
4. Defined force templates are used to incorporate military units into the joint task force.

Some of the supporting technologies are: Browsing, Interaction, Fusion, Objects, Structured Common Representation, Automatic Data Capture, and Tailoring Information To Meet User Needs. It is this last technology that our effort addresses most effectively. The USAFSAB report (15) states that:

> "The understanding of a situation or the available options depends critically on presenting the information in an appropriate form. ... the presentation must be tailored to the workflow task and to the preferences of a particular user. What is presented in the cockpit may be very different from what is presented in a command center."

As a specific example:

> "When a platoon of ground troops requests the location of enemy tanks, the JBI provides that information in a form tailored for the personal digital assistant carried in the field."

Our goal in the third phase of this effort was to define an architecture for the AMC problem domain that would comply with a JBI paradigm. We were also asked to apply the results from a previous effort (14) and to work with Maj. Robert E. Marmelstein, Deputy Chief Information Systems Division (AFRL/IFS) in formulating a working e-mail JBI paradigm.

Section 3 Building a Demonstration

Our second task was to build a demonstration including sensor visualization, terrain modeling, message traffic, large databases, and computing devices varied sizes. Our plan for performing this task was to pull together the results of numerous experiences, tools, and software. We integrated USGS databases and rendered them both in 2-D and 3-D for in-house demonstrations and for different USAF contracts. An example of a 2-D interactive Java demonstration of some of our work can be obtained by visiting our web site at www.caprarotechnologies.com. A graphic illustrating that capability is shown in figure 1. Some earlier work we performed in 3-D using Virtual Reality Modeling Language (VRML) is illustrated in figure 2. Here we integrated a Computer Aided

Design (CAD) data model of an F-15 aircraft with USGS data of the Old Forge, NY area and displayed it in 3-D. We built a simulation so that one could define the route for the aircraft to fly and one could view the scene from any vantagepoint while the aircraft is flying.

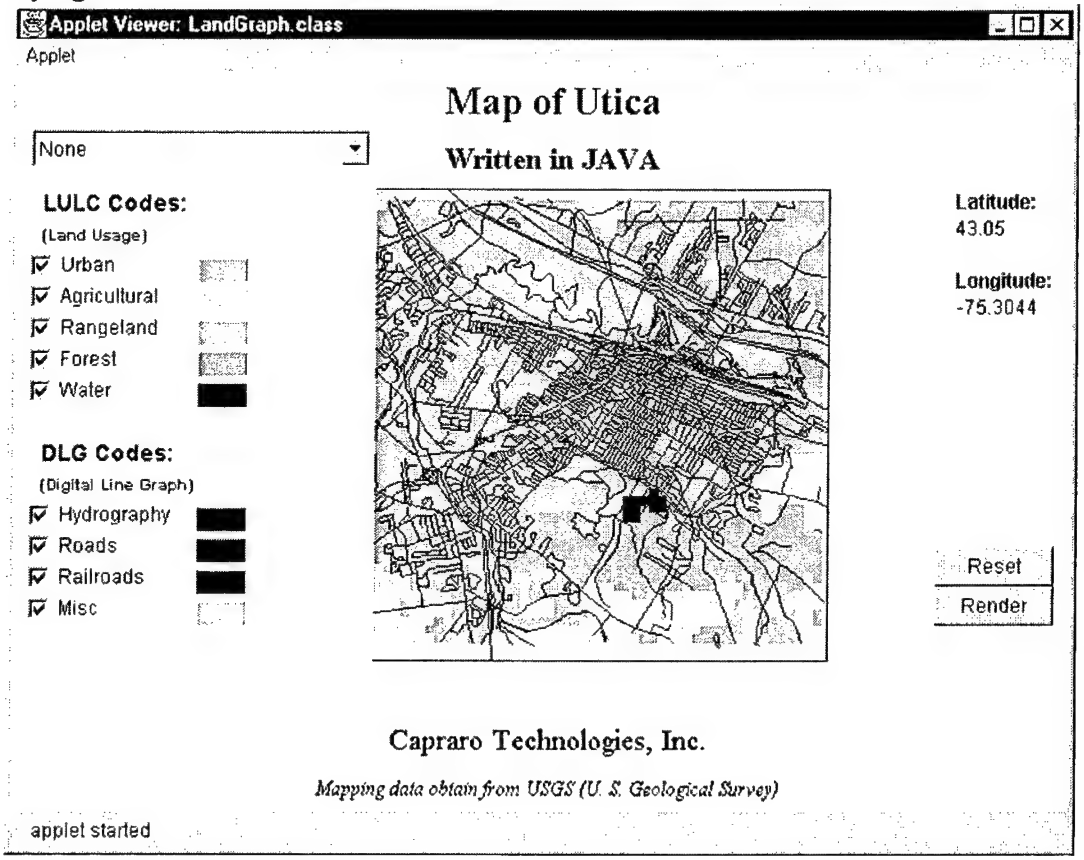


Figure 1 A 2-D USGS Rendering

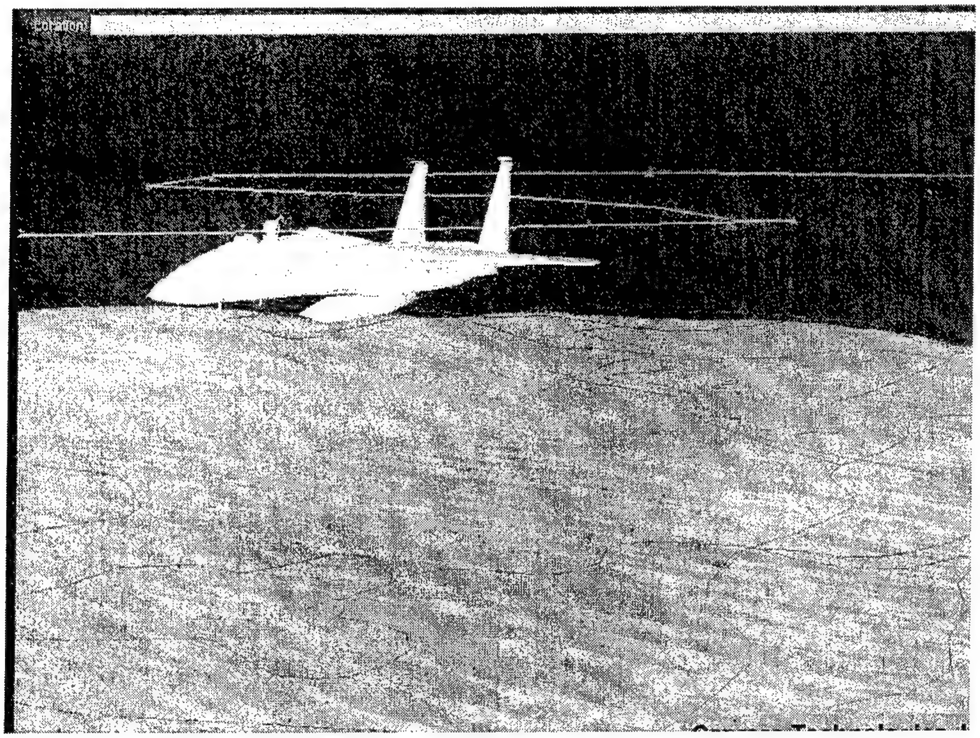


Figure 2 A 3-D F-15 & Old Forge Rendering

The previous two illustrations were built using a 3-layer software architecture model. The bottom layer consists of a relational database management system (DBMS) that contains the terrain data and objects rendered in a scene. The top layer is written in HTML, VRML, and/or Java and the middle layer is written in Java. In this manner we have tried to make our software as machine independent as possible and Internet compatible. Our design for this effort is shown in figure 3. The design currently consists of the three layers (user interface layer, control layer, and the database connectivity layer). The data compatibility layer is shown to illustrate how the databases are populated from outside data sources. This population is performed first, and then the top three layers are employed to render the data either in a browser form or in a Java 3-D or VRML display.

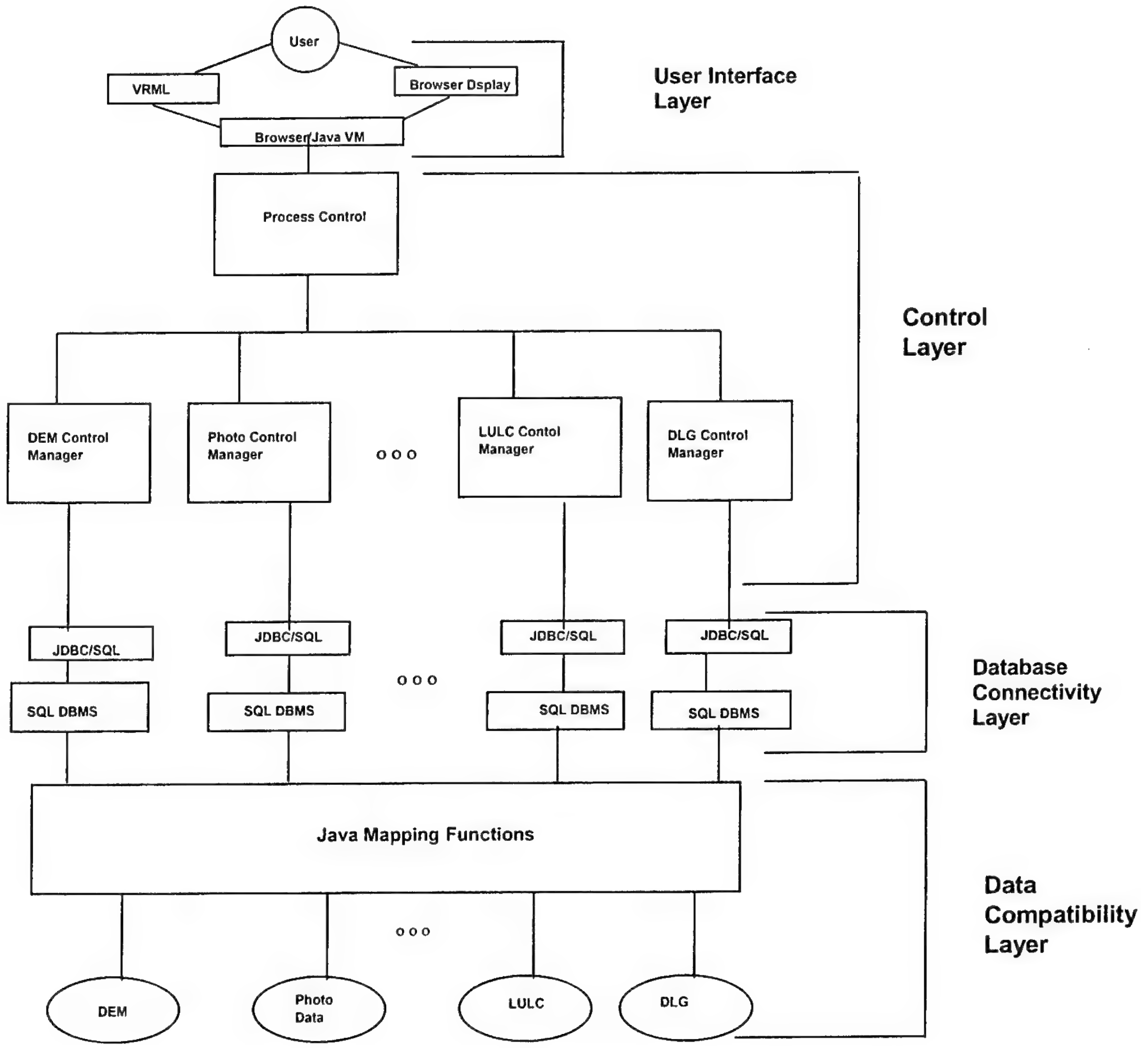


Figure 3 Preliminary Software Architecture

The current design is based upon USGS databases. These are shown at the bottom of the figure and labeled as DEM, LULC, and DLG. The Digital Elevation Model (DEM) is the USGS model of the elevations of the earth at specific intervals. Land Use and Land Cover (LULC) represents the codes for a defined piece of the earth such as urban, forest, water, etc. The Digital Line Graph (DLG) data define man-made structures such as roads, railroads, power lines, etc. and their locations. Data files for airports throughout the USA were obtained and mapping functions were written to convert these data and populate a relational DBMS. These functions were written in Java and populated a database using Microsoft SQL Server DBMS. A 2-D Java view of this data for the Eglin AFB is shown in figure 4. The DEM data were processed also and a photographic image of the Eglin AFB was obtained. Multiple Java development environments and 3-D tools for rendering the resultant scenes were evaluated. Parallel Graphics' Cortona VRML Viewer was chosen for rendering 3-D images. The viewing software is free and can be obtained from the Internet. A link to their site is provided at our web site. See figure 5.

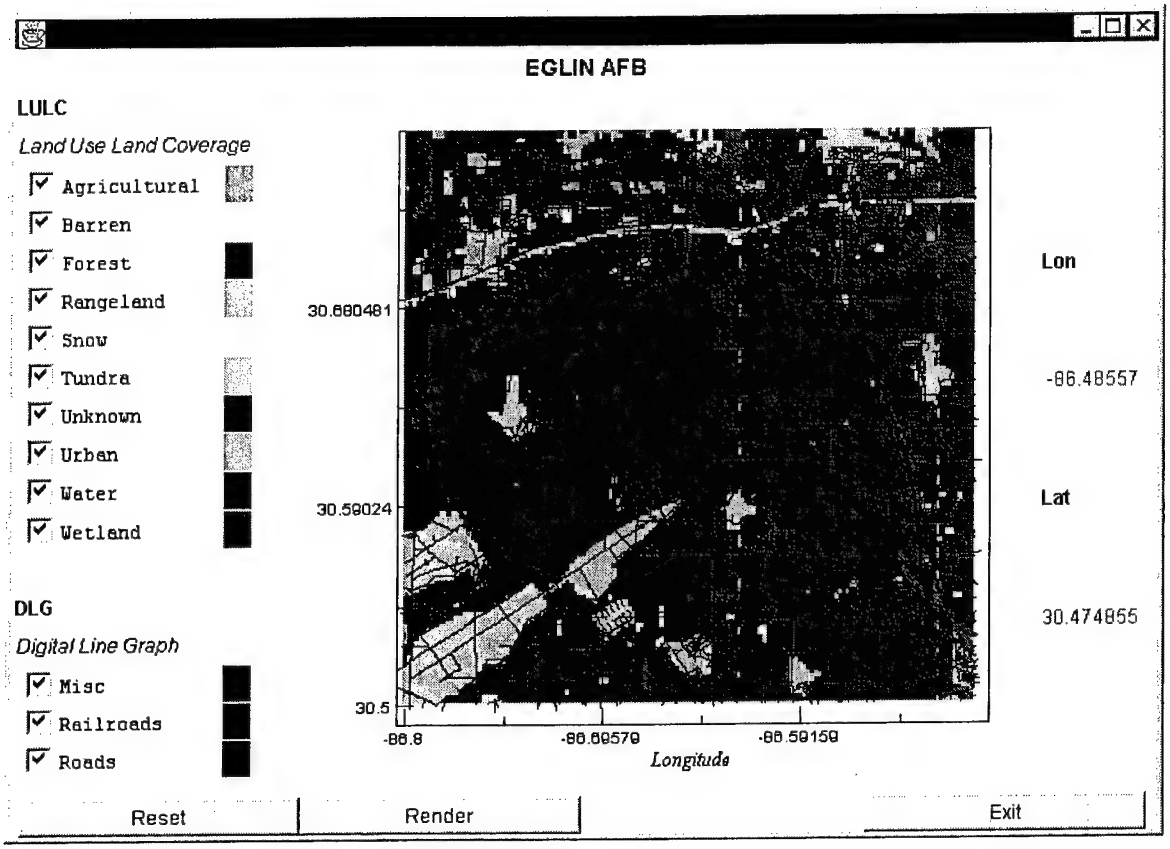


Figure 4 A 2-D Eglin AFB Rendering

The data compatibility and connectivity layers (see figure 3) are complete and the Control and User-Interface layers are partially complete. Instead of finishing a full design of all of the layers we have moved our architecture design for the AMC domain to a JBI instantiation. With that as our new goal, for the next task we leveraged a previous effort's results and implemented a demonstration on our web site. For complete description of the web site please see (14). From the "Flight Query Demo" shown in figure 5 if the user first downloads the VRML viewer then one can download and view one of two 3-D renderings of the Eglin AFB area. See figure 6. The first is viewed by depressing the button labeled "VRML Demonstration of Eglin AFB" and then "VRML Eglin AFB". A series of views that one could create are shown in figures 7-10. If the ultimate user is at a location where bandwith or processing capability is limited then one could send one or two of these views rather than the VRML software. If bandwidth were not an issue then the user could send the total file and the pilot could fly through the scene. If however a pilot was going to land at an airport that had some bomb damage someone could create the same 3-D rendering with photographs obtained from an intelligence agency satellite of the area. We distorted the original web site image obtained from the USGS and added simulated bomb damage. See figures 11 – 14. We also added some buildings to show that detailed models of the area could be captured as well. The simulated battle damaged scene can be obtained by choosing the "VRML Eglin AFB (Damage) option on the above mentioned web page.

A demonstration of this full capability has been provided to the USAF. The web site is resident on a CTI web server and can be accessed by USAF personnel for demonstration purposes.

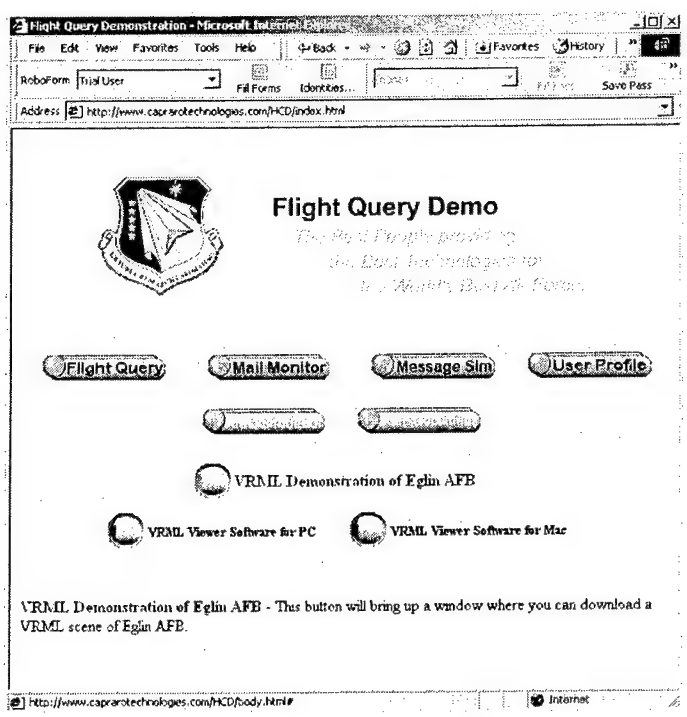
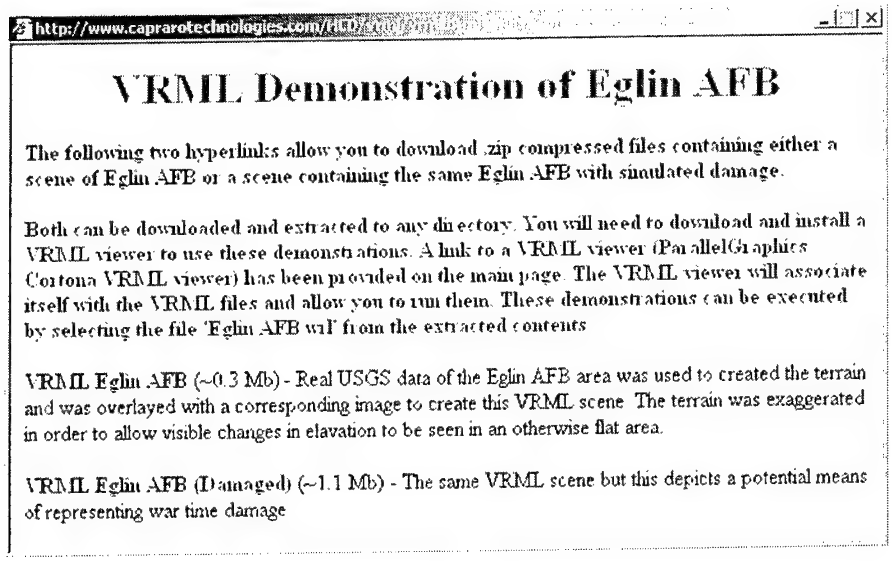


Figure 5 Website Switchboard Page



http://www.caprarotechnologies.com/HCD/

VRML Demonstration of Eglin AFB

The following two hyperlinks allow you to download .zip compressed files containing either a scene of Eglin AFB or a scene containing the same Eglin AFB with simulated damage.

Both can be downloaded and extracted to any directory. You will need to download and install a VRML viewer to use these demonstrations. A link to a VRML viewer (ParallelGraphics Cortona VRML viewer) has been provided on the main page. The VRML viewer will associate itself with the VRML files and allow you to run them. These demonstrations can be executed by selecting the file 'Eglin AFB.wrl' from the extracted contents

VRML Eglin AFB (~0.3 Mb) - Real USGS data of the Eglin AFB area was used to created the terrain and was overlayed with a corresponding image to create this VRML scene The terrain was exaggerated in order to allow visible changes in elavation to be seen in an otherwise flat area.

VRML Eglin AFB (Damaged) (~1.1 Mb) - The same VRML scene but this depicts a potential means of representing war time damage

Figure 6 VRML Selection Page

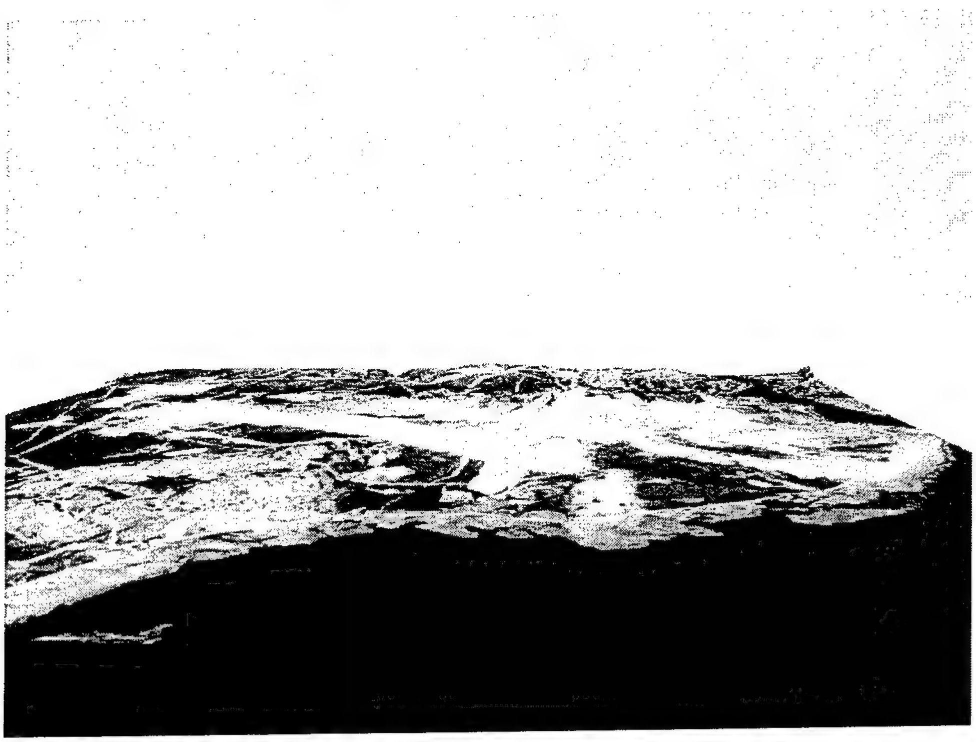


Figure 7 Eglin

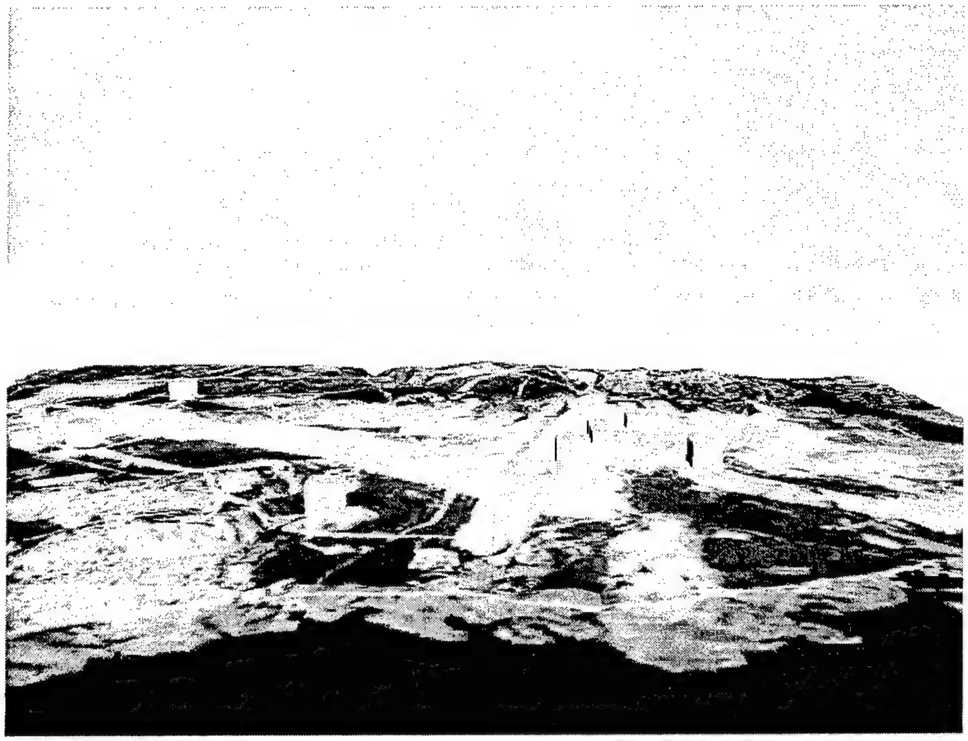


Figure 8 Eglin Simulated Buildings

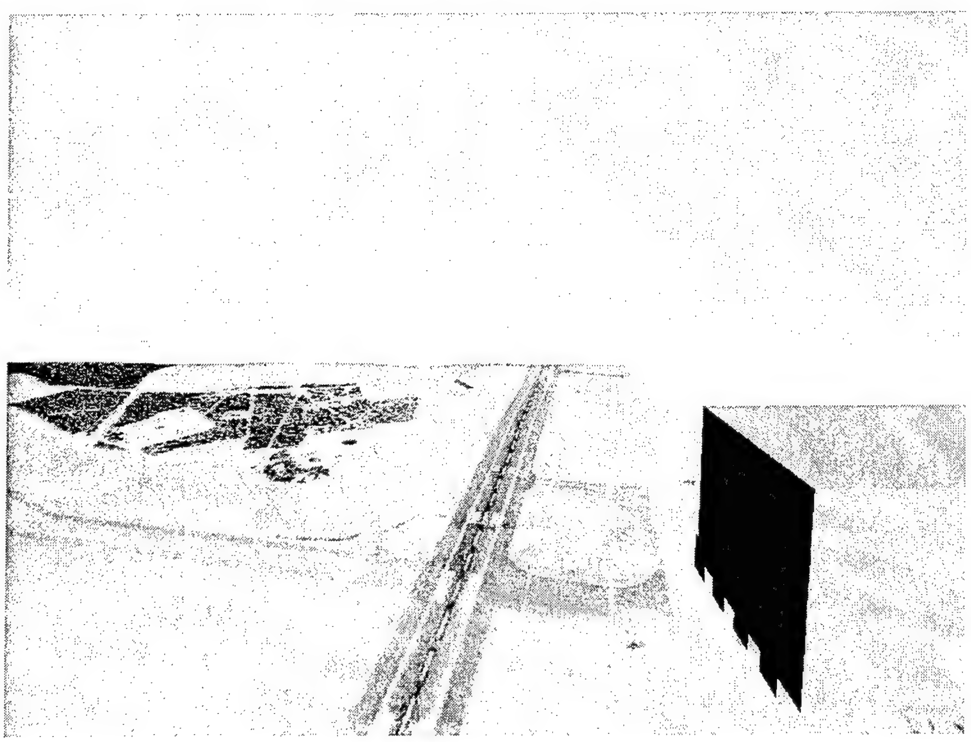


Figure 9 Eglin Runway Approach

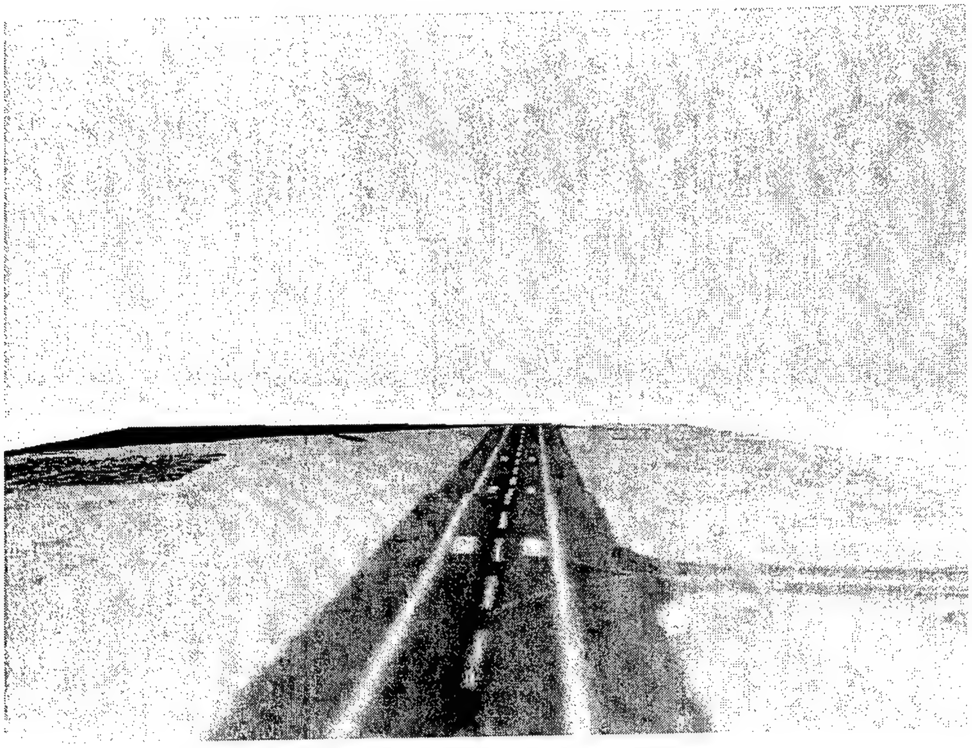


Figure 10 Eglin Runway

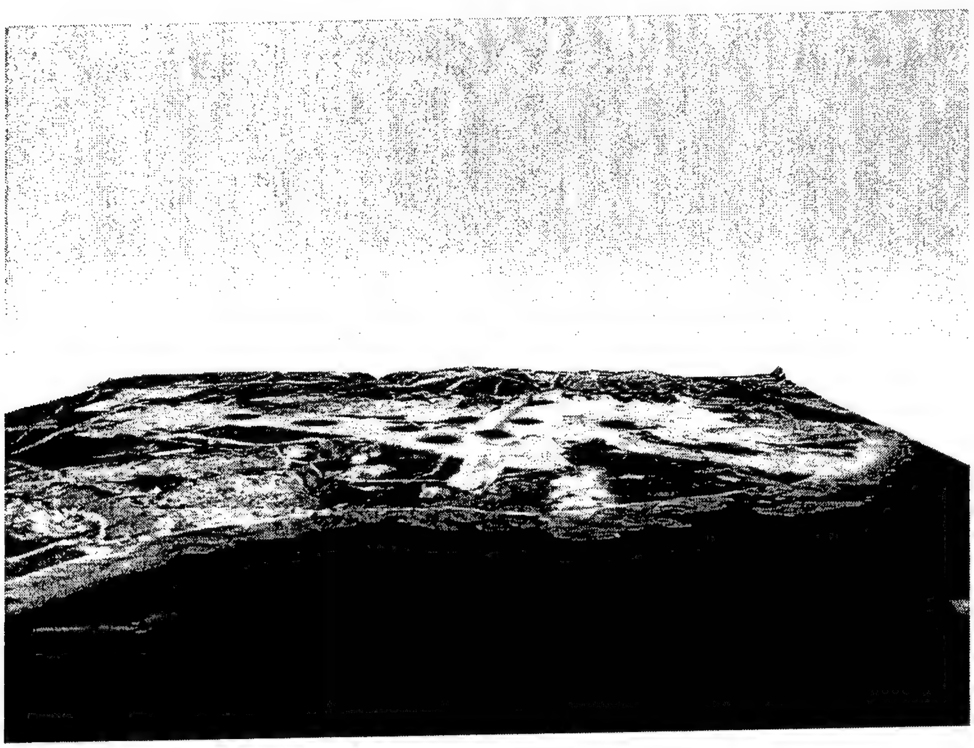


Figure 11 Eglin Damage

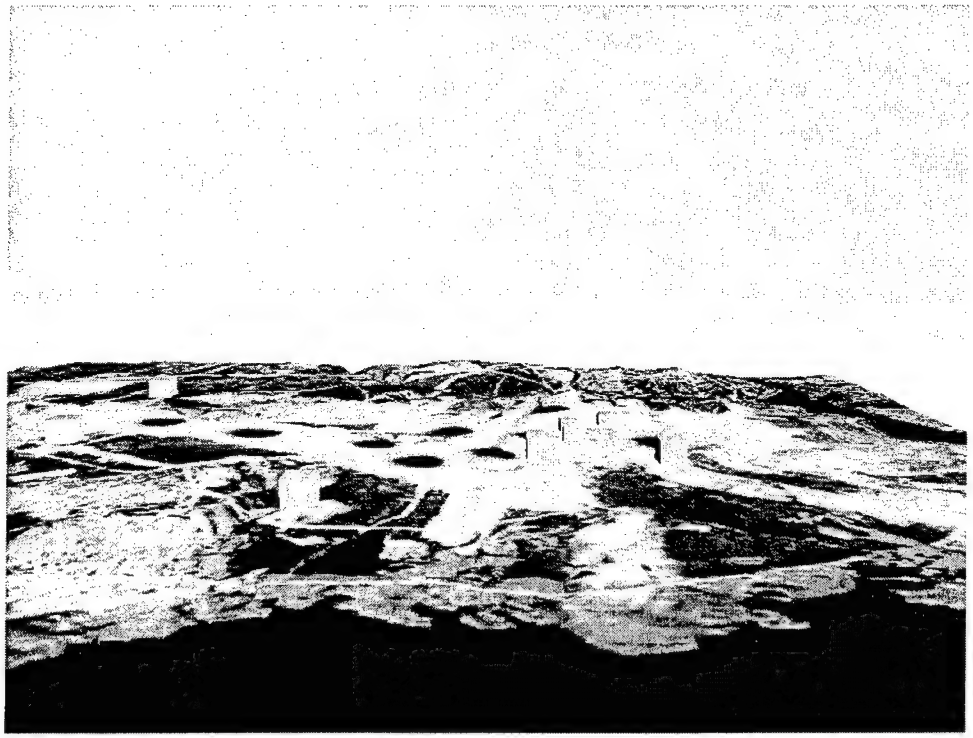


Figure 12 Eglin Damage Simulated Buildings

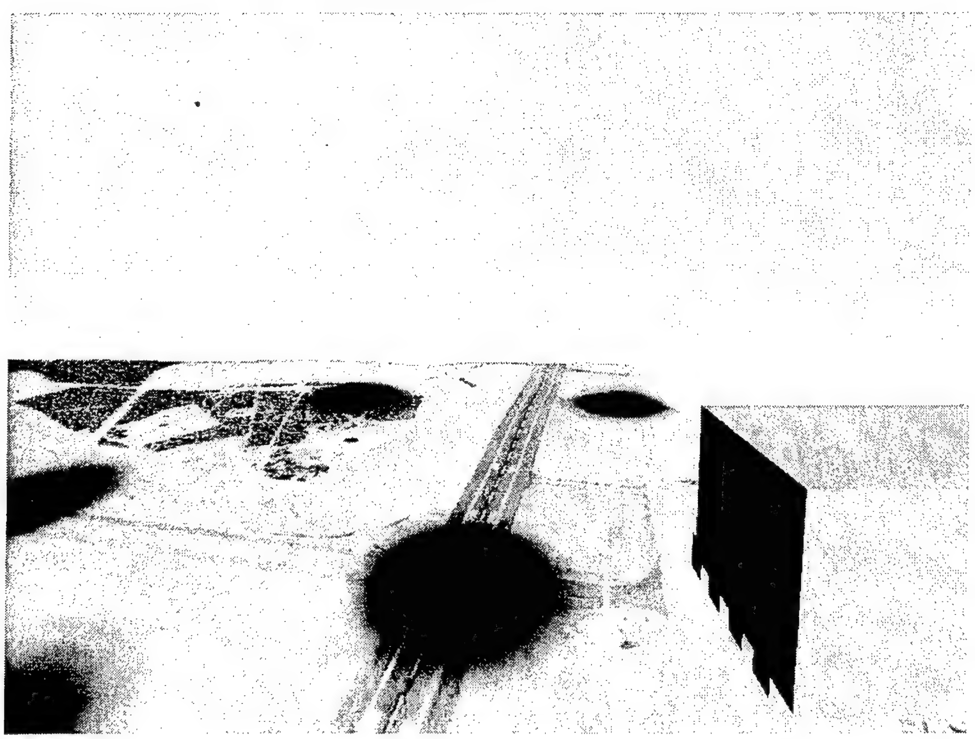


Figure 13 Eglin Damage Runway Approach

Figure 14 Eglin Damage Runway

Section 4 Demonstration Analysis

Screen shots of the demonstration are shown in figures 7 - 14. The demonstration illustrated that commercial software tools are available to integrate USGS DEM data and warp images to build VRML models. These models could then be used to simulate flying into these airports. In so doing they could be sent as a VRML model to a distant location for a pilot to exercise before taking off, if they had the capability of retrieving data files of 600 Kbytes. If they have limited bandwidth, then someone at headquarters could simulate flying the approaches, capture one or two still images, compress the images and then send them to the pilot using bandwidth deprived communications links using IFGR. However, building an architecture to satisfy this need at AMC may not be the best solution, since it will be replicating functions performed elsewhere. Another organization, either NIMA or the intelligence community could gather and warp images to elevation terrain data obtained from the intelligence community and NIMA. NIMA already provides the approach plates to AMC and it develops similar terrain data as the USGS, but worldwide, not just the USA. NIMA or the intelligence community would have to acquire each other's data to integrate and create the VRML files for any area in question. In this regard we suggest that a JBI paradigm be investigated. Consider the following high level design shown in figure 15.

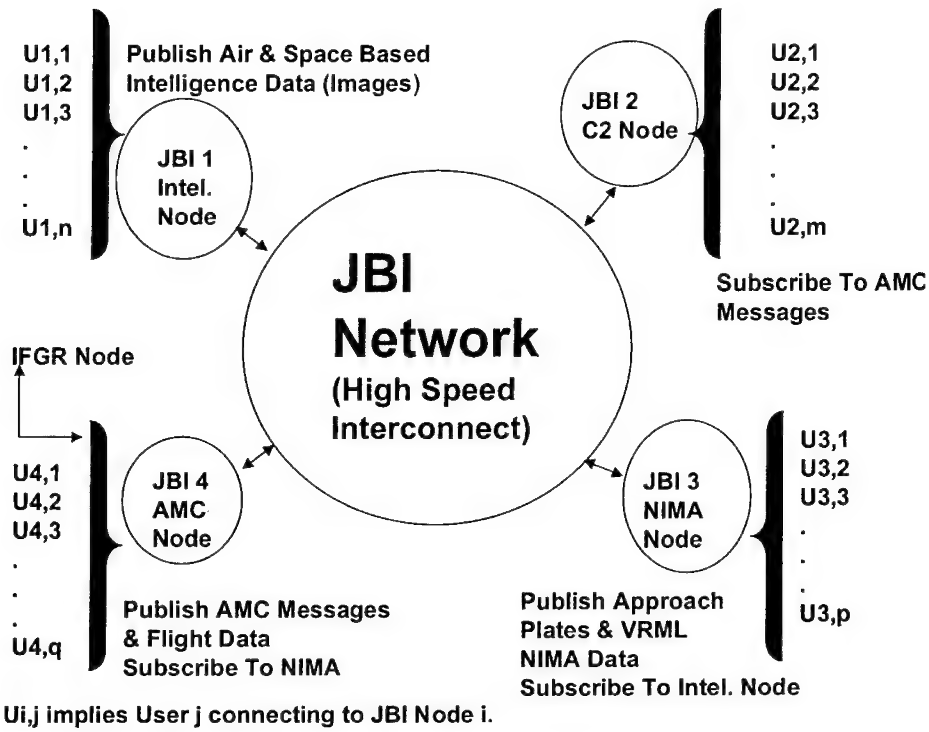


Figure 15. A JBI Architecture Solution

The architecture shown is based upon the JBI architecture (15,16) where the JBI Network represents a high speed connection to multiple JBI nodes. Each node is made up of multiple computers and servers that are providing information to the network through publications of material. These materials may be reports, videos, databases, etc. They also receive and maintain subscription requests from users, shown here as Ui, j. These are individuals or processes that are entering the JBI Network via their computing devices through high and low speed bandwidth connections.

Shown in this hypothetical architecture are four nodes. JBI 1 is a node representing the intelligence community that publishes images taken form air and space based platforms. These images may be of airports, troop movements, battle damaged areas, etc., gathered from different types of sensors ranging from radio frequency (RF), to infrared (IR), to optical.

JBI 2 represents a command and control (C2) node. It may be the command post located near a battle area. As shown here, it may be subscribing to AMC's messages. This command post is interested in what flights have taken off, what cargo is on board, when they expect to arrive, etc. The figure does not show that this node is publishing any documents. In reality they would most likely be publishing reports related to troop movements, battle plans, etc.

JBI 3 represents a node at NIMA. For this discussion this node will subscribe to image publications at the intelligence node and it will publish VRML software for different parts of the world, see Airfield Initiative Program (16). It is this node that will gather the

elevation data for a particular area of the earth, warp an image, retrieved from Node JBI 1, to the elevation model, and create the VRML code for its multiple subscribers throughout the JBI Network.

JBI 4 is a node at AMC. This node publishes the AMC messages and flight data concerning all its flights and it also subscribes to NIMA for the gathering of VRML code for different airports and makeshift air strips that can appear anywhere in the world. Its users are different AMC commanders and pilots requesting information about flight approach plates and detail information about air strips. Once the data are obtained then JBI 4 can either provide the information via its local network to its users or if a user were on an aircraft or remote, bandwidth deprived site, the information would use the IFGR to send the information to the user.

The user can get information via a push or pull approach. In the example shown above a user logs onto a web page and pulls the information from the site. If the user subscribed to the AMC messages published at the AMC JBI 4 node, then information regarding these messages could be pushed to the user when the information they require is available. Both the push and pull approaches using simulated AMC message types was demonstrated within a previous effort (14).

A hypothetical scenario that this architecture can support is presented in the following steps:

1. A request from the battlefield is made to the C2 Node that a transport is needed to evacuate wounded soldiers.
2. A query is made to the AMC Node to see if there are any transports near the area in question.
3. A request is made to AMC for assistance.
4. A study of the aircraft location, cargo, and crew is performed and a flight is identified for diversion.
5. An IFGR diversion request is sent to the pilot. In parallel AMC requests a VRML code for the airstrip in question from NIMA.
6. NIMA queries for a recent image of the area if it does not have one and creates a VRML code for AMC.
7. Once the VRML code is received at AMC, a pilot uses the software to simulate different approaches and selects one or two images to depict the best approach.
8. The image is attached to a message describing the necessary information for landing at the airstrip (e.g. weather information) and attaches the image to the message. The message is then sent via IFGR to the pilot.

It is our recommendation that a JBI architecture is well suited for providing immediate, accurate, and detailed information that are required by the USAF and AMC. The technology is there to support their needs. As the infrastructure of a JBI becomes available and understood the solutions to scenarios similar to the above will be intuitive in their implementation.

Section 5 JBI Study and Role within AMC Domain

As was mentioned earlier, a task was added to this effort to study the JBI paradigm and evaluate the models completed in task 3 and determine their role in meeting JBI requirements. In the design shown above each user is connected to the JBI with different devices and different bandwidths. Consider the following diagram in figure 16. From remote and/or intelligent devices a user communicates with the JBI node to enter the JBI network. In a previous effort (14) an AMC message simulator and a proof of concept instantiation was built of the Intelligent Interface as shown in figure 16. A demonstration was performed showing how a user could gather information from a web site using a pull process, change their user profile, and obtain information from a site irregardless of the user's device. The Intelligent Interface software provides information to the user, appropriately dependent upon their device. The user could set a profile to receive information from the AMC messages when specific occurrences occurred. This demonstrated a push capability. When particular events occurred, the intelligent software would send the user an email with the data requested. The system also allows a user to query the system by sending email queries using hand-held computing devices (HCD), e.g. a Palm Pilot or Windows CE device. The intelligent software monitors the mailbox, parses the message and performs one of three queries, packages the results, and sends an email back with the desired information.

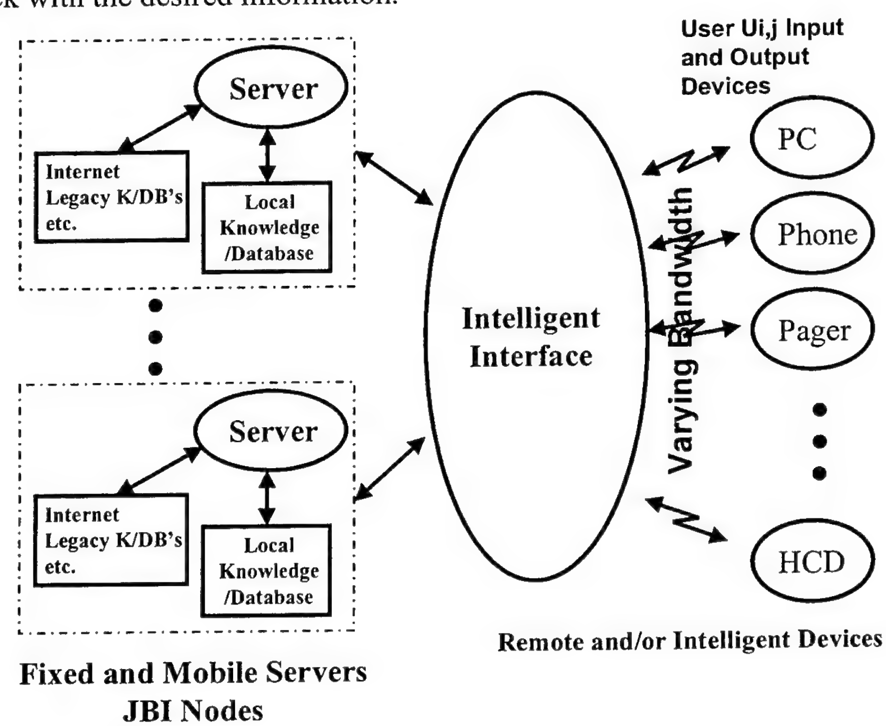


Figure 16. Intelligent User Interface

This capability would allow a user to create and maintain subscriptions to a JBI node using an email paradigm. Major Robert Marmelstein of the AFRL/IFS Division was independently investigating email as a method for implementing a JBI publish and subscribe (P/S) paradigm. Integrating our approach with his seemed natural. We would

extend our capability by employing a JBI P/S paradigm and provide Maj. Marmelstein with a node to handle his subscription requests. This was a great opportunity to investigate a JBI email P/S intelligent interface design.

To marry our previous work with an email P/S paradigm we needed to work out numerous additions and changes to our current system. We had approximately three meetings with Major Marmelstein and exchanged many emails. First we had to find what we were going to publish and then determine how Major Marmelstein wanted to perform his subscription requests. Since his application was totally built using Microsoft products and tools and ours are built using Java and SQL running on Linux, we could not share software. However, this exercise demonstrated that we could define our interfaces using open system W3C standards and integrate data within these heterogeneous systems.

We decided that we would publish the status of flights as if we were the AMC node. In this way we could reuse our flight simulator that generates messages for take-offs, landings, and auto position reports for flights. In this effort we wanted to exercise some of the major aspects of the JBI. We demonstrated both push and pull of information depending upon a user's profile and the dynamics of the data. We demonstrated, in some small way, the publish and subscribe paradigm. We showed that we were dynamic in our capability to change processes and control mechanisms in real-time. Lastly, we showed an interface to fuslets as defined within the JBI reports. Noting that we could not use the actual data that are contained within the AMC database, we needed to create a capability that would simulate the messages that would be generated within their system.

To understand our previous effort and how we used its results in this effort, a brief review of (14) is required. The user creates the flights by choosing the day, up to five flights, assigning a mission number for each flight and selecting the ICAO code of departure and destination. Given these data and the aircraft's speed, the program will then generate the departure message, the 15-minute position messages in route, and the landing and arrival gate message. For demonstration purposes once the user chooses which day they wish to simulate, another Java code searches the database and generates the messages and stores them into another database as if they were messages received from an actual aircraft for the day chosen. A user acquires this opportunity when they first enter the demonstration web page (see figure 5).

Once the user enters the switchboard page they can maneuver through the demonstration pages and functions. For moving to the message simulator page, the user would click on the third button from the right, labeled Message Sim. Once activated, a page similar to that shown in figure 17 will appear.

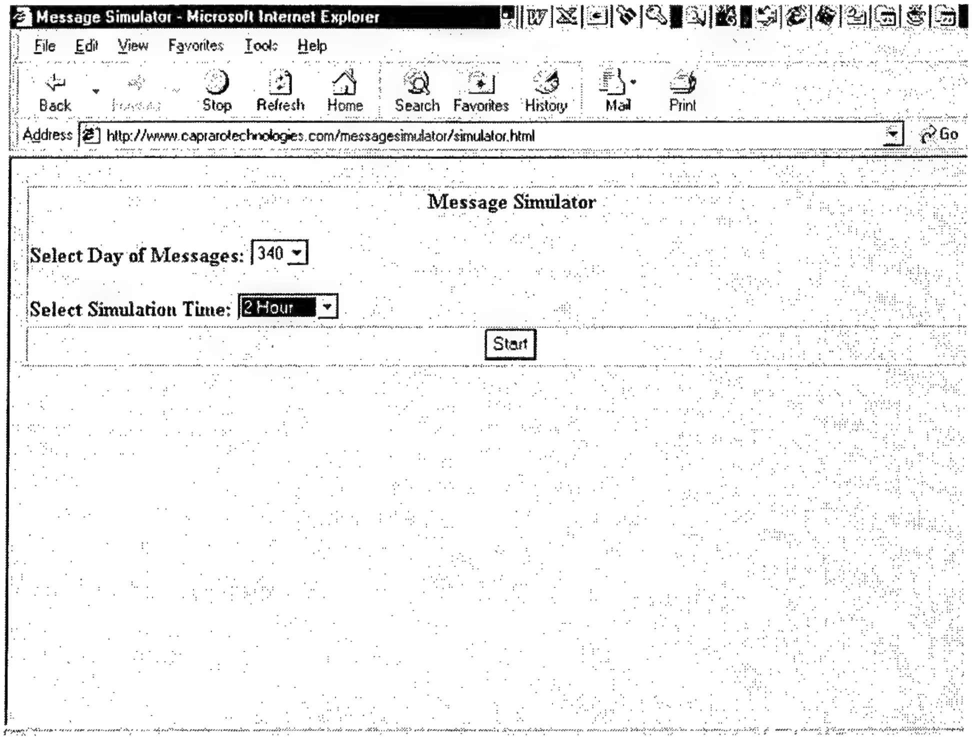


Figure 17. Simulator Page

The user can choose one of five Julian days to simulate, numbered 340 through 344. Next, the user can choose whether to run the simulation for a total of 24 hours, over a simulated period of 0, 1, 2, 3 or 4 hours. The first choice of 0 hours was provided for diagnostic checking of our software. If the user wants to provide the demonstration over a short period of time they can choose the 1-hour option.

The demonstration displays results based upon the user's profile that can be changed in real-time, meaning that the system will respond to the user's change in profile immediately after the change is recorded within the meta-database. The user profile page is accessed from the main switchboard (figure 5) by activating the button on the far right. Once activated, a current list of all the users is shown and the capability of adding a new user is provided. The current design only allows a user to enter name, email address, and type of device.

The user can choose whether they wish to have the status of flights provided to them periodically or only when triggered by specific events. If the user chose to have email messages sent to them triggered by an event, then the two options now available are either a departure or an arrival from a particular country. For example, if a user chose "Event" equal to departure and "Location" equal to United States, then any time an aircraft departed from the US, a message status of all flights for that day would be sent, similar to the message shown in figure 18. The user can also request that no messages be sent to them ("Event" equal to "No Notification") and receive flight status information by pulling the data from the site themselves.

Flight Number: MC0503
Departure Time: 1630
Departure Location: K63G (Chicago / Calumet Coast Guard Station)
Destination Location: KLAX (Los Angeles, Los Angeles International Airport)
Last Reported Time: 1856
Flight Arrived at: 1856
Reached Terminal at: 1910

Flight Number: MC0603
Departure Time: 1045
Departure Location: EDDG (Muenster / Osnabrueck)
Destination Location: TISX (Christiansted / Alex. Hamilton Field, Saint Croix)
Last Reported Time: 1653
Flight Arrived at: 1653
Reached Terminal at: 1700

Flight Number: MC0703
Departure Time: 1755
Departure Location: EGXH (Honington Royal Air Force Base)
Destination Location: KMXF (Maxwell Air Force Base / Montgomery)
Last Reported Time: 2055
Last Reported Position Coordinates:
49 Deg 47 Min N Latitude
56 Deg 16 Min W Longitude
Percent of Flight Complete: 55.32

Flight Number: MC0802
Flight Departed Yesterday
Departure Location: OIZJ (Jask)
Destination Location: BIX2 (Biloxi, Keesler Air Force Base, Navu)
Last Reported Time: 0015
Flight Arrived at: 0015
Reached Terminal at: 0023

Flight Number: MC0803
Departure Time: 1215
Departure Location: BIX2 (Biloxi, Keesler Air Force Base, Navu)
Destination Location: UHPP (Petropavlovsk-Kamchatskij)
Last Reported Time: 1927
Flight Arrived at: 1927
Reached Terminal at: 1937

Flight Number: MC0903
Departure Time: 1630
Departure Location: FZEA (Mbandaka)
Destination Location: RCDC (Pingtung South Air Force Base)
Last Reported Time: 2045
Last Reported Position Coordinates:
16 Deg 28 Min N Latitude
61 Deg 54 Min E Longitude
Percent of Flight Complete: 45.47

Figure 18. Typical email Message

To pull information from the site, the user logs onto the site and from the switchboard page shown in figure 5, the user would then choose the Flight Query button. This action will bring up a web page for the user to formally log into the system for retrieving information (figure 19). The user will enter first and last name and activate one of three options: Map Flights, Time Line, or Reset Values. The last option will allow them to re-enter a correct version of their name if it was misspelled. If the user chooses Time Line, the system will return a page similar to that shown in figure 20, where each row represents either a flight departed that day or a flight departed but not landed the day before. This picture allows the user to view the respective time lines of all flights for that day based on the current time. The mission numbers of each flight are shown along with the current time, indicated by the green vertical line.

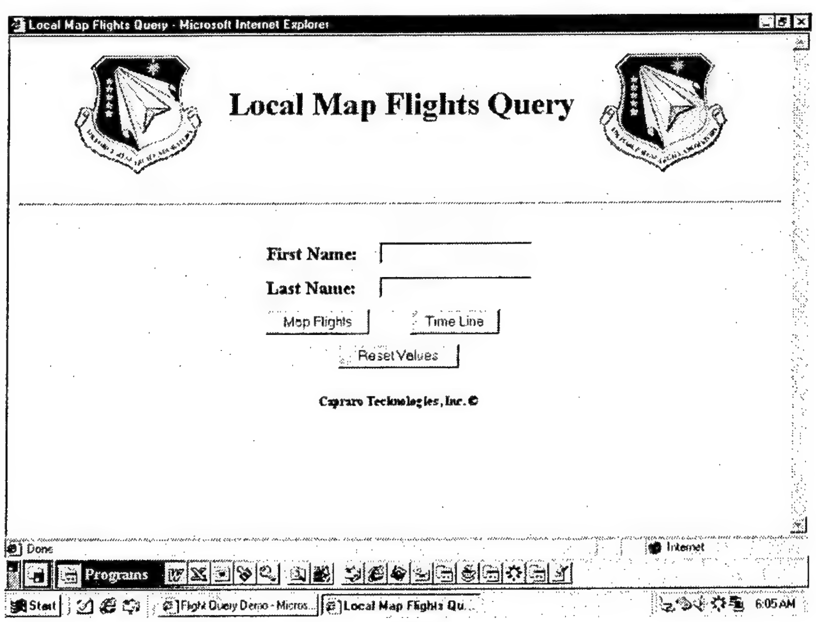


Figure 19 Flight Query Page

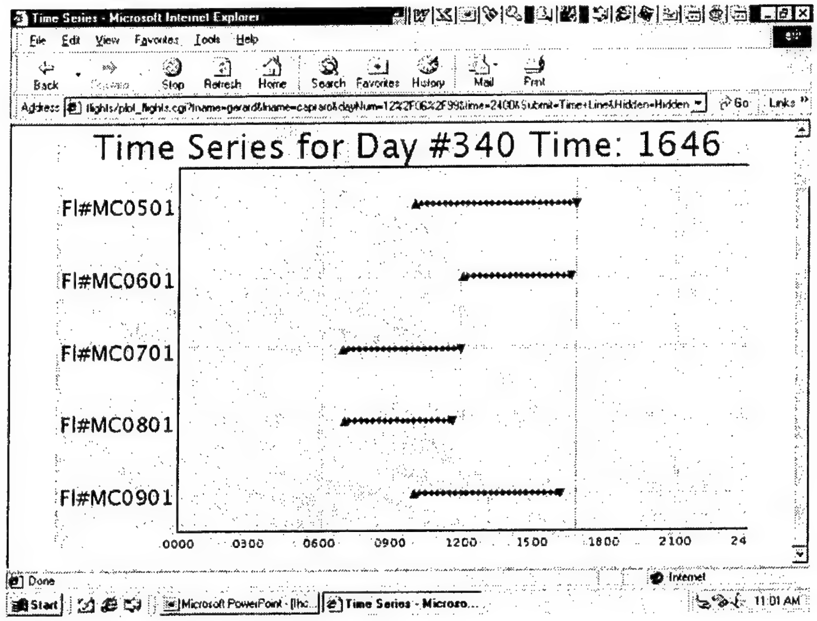


Figure 20 Time Line Page

If the user would like more detail on each flight they can choose the Map Flights button shown in figure 19. This action will provide the user with a world map highlighting each of the flights that were active for that day with their latest position (see figure 21). Each flight is shown using a different color. The user can obtain more information about each flight by clicking on any one of the paths shown on the map. This action will display, for example, the information shown in figure 22. From this page the user can choose one of two buttons to activate. These actions will provide additional information related to the crew or the passengers as shown in figures 23 and 24 respectively.

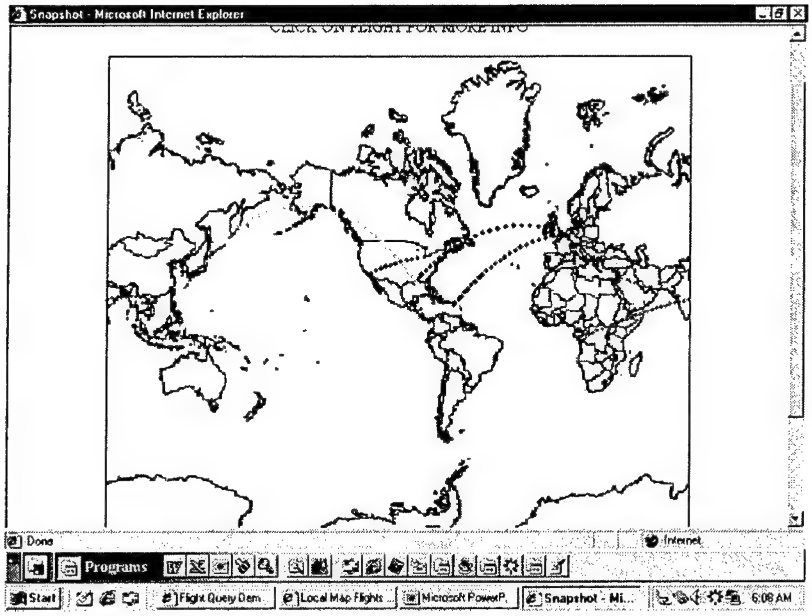


Figure 21 Flight Map Page

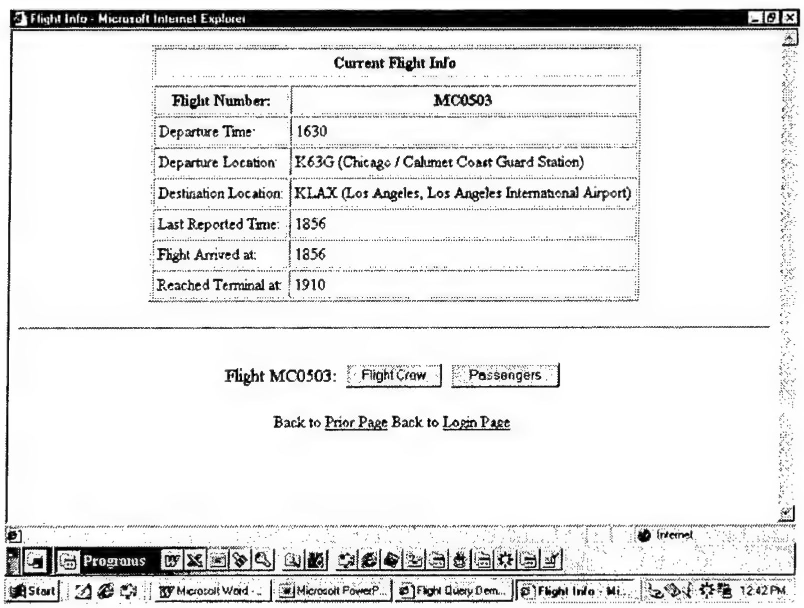


Figure 22 Flight Information

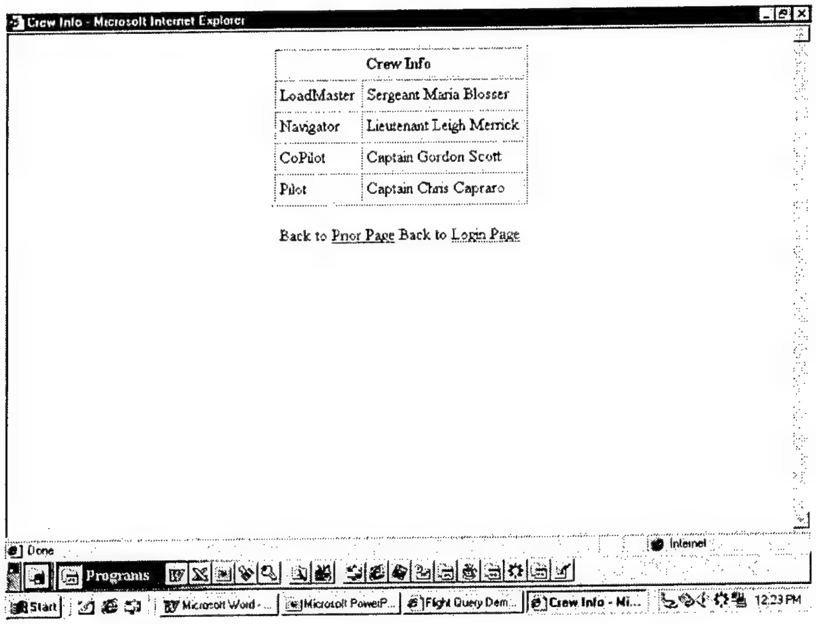


Figure 23 Crew Information

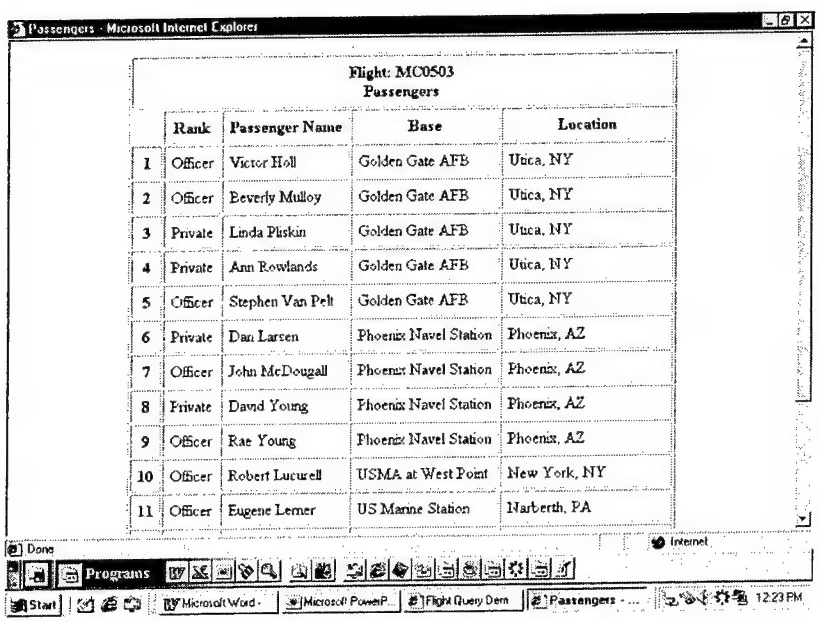


Figure 24 Passenger Information

If the user enters the system with a device that has a browser front end but is limited in screen capability (e.g. Windows CE handheld device) or would rather have the information provided in text form, then one can enter a first and last name as "h" and

“hcd” respectively (see figure 19). Then if one activates either the Time Line or Map Flights the user will get a response similar to what is shown in figure 18.

In addition to pushing information using email and pulling information using web enabled devices, we can also provide a user the capability to pull information using email. This works well whether the user is pulling the information with a Palm device or a desktop computer. To activate this capability the demonstrator clicks on the Mail Monitor button shown in figure 5. Once activated a user can send an email to a special mailbox on our site with one of three entries for the subject of the email. These are Flight mcxxxx, or Passengers mcxxxx, or Crew mcxxxx. The user substitutes the last four digits of the flight number of interest (xxxx). Once our mail server receives the message, it is retrieved by a Java mail monitoring program. Another Java program saves the sender's email address, parses the subject line, queries the database and sends the query results back to the requester. The following three figures are representative responses to requests for Flight mc0503, Passengers mc0803, and Crew mc0803.

Flight Number: MC0503
Departure Time: 1630
Departure Location: K63G (Chicago / Calumet Coast Guard Station)
Destination Location: KLAX (Los Angeles, Los Angeles International Airport)
Last Reported Time: 1856
Flight Arrived at: 1856
Reached Terminal at: 1910

Figure 25 email Flight Request

Passengers on Flight: mc0803

0. Private Jennifer Johnson
Phoenix Navel Station
Phoenix, AZ

1. Private Roger Septoff
USMA at West Point
New York, NY

2. Officer Raymond Steinberg
US Coast Gurad Station
Southfield, MI

3. Officer Joseph Riscili
USAF Flying Academy
Amherst, NY

4. Officer David Hanson
Canada Air Station
Mississauga, Ontario

5. Officer Gordon Lamb
Royal Marine Station
Tyne & Wear, Newcastle

6. Private Jim Campbell
McDonald's Air Force Station
Glasgow, null

Figure 26 email Passengers Request

Crew For Flight: mc0803

Navigator: Lieutenant John Gross
Baltimore Navel Yard
Baltimore, MD

LoadMaster: Sergeant Gary Johnson
Kauai Air Station
Kauai, HI

CoPilot: Captain Leonard Croth
Canada Air Station
Mississauga, Ontario

Pilot: Captain Renee Capraro
Fort Hill
Camp Hill, PA

Figure 27 email Crew Request

The above description is a brief overview of what we had accomplished in our previous effort. In this effort we proposed to leverage the W3C technologies such as the extensible markup language (XML) and interface this technology with a JBI email paradigm. In so doing we had to decide what information we could generate that would be a "realistic" document that we could publish that would make sense for Major Marmelstein to subscribe to and would fit within our JBI architecture solution as shown in figure 15. We decided on using only the flight status data as illustrated in figure 18. We envisioned that AMC would publish many documents such as maps (e.g. figure 20), timelines (e.g. figure 21), etc. However we chose a simpler XML document depicting two columns of data, one column showing those flights per day that are currently in flight

and the other showing those that have landed. A typical XML document displayed in an XSL style sheet is shown in figure 28.
The same simulator that generates the flights for a particular day kicks off a Java routine that updates the XML document based upon the messages generated.

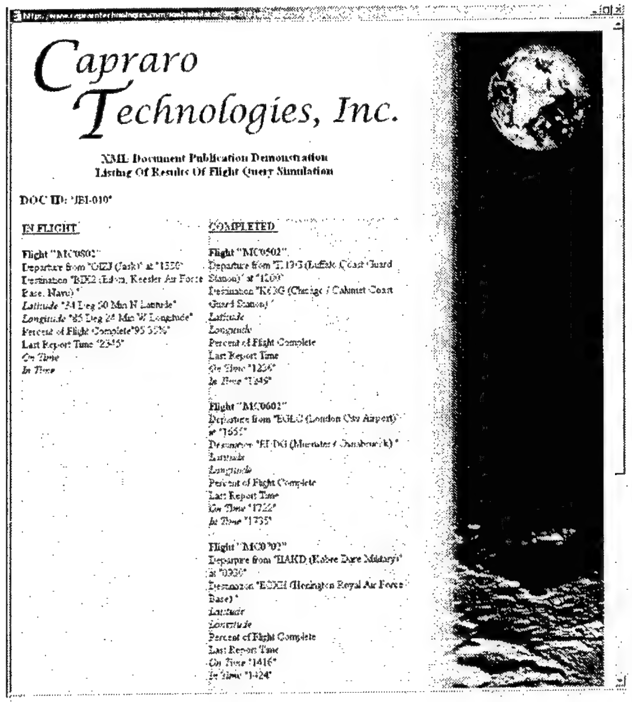


Figure 28 Typical AML/XSL Document

We then needed to work with Major Marmelstein to adjust our software to interface with his existing software. His software was going to send us XML documents to subscribe for portions of the above described publication, i.e. JBI-010. The attributes/entities/elements we worked out together that a user would use in specifying a subscription request are provided in table 1 and an example subscription request is shown in figure 29.

Attribute/Entity/Element	**Contents**	**Description**
JBI_SUBREQ	The XML subscription request.	Header for the XML subscription request.
EM_ADDR	An email address	This is the address of the person or process submitting the subscription request.
USER_ID	A user identifier	This is a unique identifier agreed upon between the publisher and subscriber for the individual. It may also be a unique identifier

		assigned by the JBI.
RES_ID	A unique identifier for the information resource that is being published	This is a unique identifier for the information resource that is either determined by the publisher or the JBI.
STREAM_ID	A unique identifier for this subscription.	A unique identifier generated by the subscriber. Note there may be more than one subscription, to the same resource, from the same user, and active at the same time. This ID makes each of them unique.
ACTION	ACTIVE, SUSPEND, RESUME, or EXPUNGE	The subscriber can activate a subscription. Once activated a user can then suspend it or expunge it. If it suspended then the user can resume it if it hasn't run out of time or has been expunged.
Recurring	True or False	If true then determines that the request should be done more than once. If false this is a one time request.
TimePeriod	True or False	If true then a start and end date of the subscription will follow. If false then no start and end date will follow.
StartDate	MM/DD/YY HH:MM	The month, day, year, hour and minute to begin the subscription.
EndDate	MM/DD/YY HH:MM	The month, day, year, hour and minute to end the subscription.
RefreshRate	Seconds	The value in time of the interval when the publisher will push a document to the subscriber. If the value is zero then the publisher pushes a document to the subscriber when ever there is an update.
SubsetValue	True or False	Determines whether the subscriber wishes the total document, i.e. False or a

		subset of the document, i.e. True.
Method	XPATH or XSL	Determines the method the subset of the document is generated.
SubQuery	XPATH query string	The XPATH query string that acquires the subset of the published document.
SubQueryFile	XSL path or file name	The XSL path or file name that acquires the subset of the published document.

Table 1. Subscription Request Description


```
<JBI_SUBREQ>
    <EM_ADDR>robert.marmelstein@rl.af.mil</EM_ADDR>
    <USER_ID>rmarmels</USER_ID>
    <RES_ID>JBI-010</RES_ID>
    <STREAM_ID>s001</STREAM_ID>
    <ACTION>ACTIVATE</ACTION>
    <Recurring value = "TRUE">
        <TimePeriod value="TRUE">
            <StartDate  value="10/29/2000 16:00" />
            <EndDate   value="11/20/2000 08:00" />
        </TimePeriod>
        <RefreshRate value="3600" />
    </Recurring>
    <Subset value="TRUE" method="XPATH">
        <SubQuery>//Flight[FlightNo='MC0501']</SubQuery>
        <SubQueryFile />
    </Subset>
</JBI_SUBREQ>
```


Figure 29 Example Subscription

The attributes/entities/elements we worked out together that a user would use in acknowledging a subscription request are provided in table 2 and an example acknowledgement is shown in figure 30.

Attribute/Entity/Element	**Contents**	**Description**
JBI_SUBACQ	The XML subscription acknowledgement.	The header for the XML acknowledgement
RES_ID	A unique identifier for the information resource that is being published	This is a unique identifier for the information resource that is either determined by

		the publisher or the JBI.
STREAM_ID	A unique identifier for this subscription.	A unique identifier generated by the subscriber. Note there may be more than one subscription, to the same resource, from the same user, and active at the same time. This ID makes each of them unique.
STATUS	ACTIVE, INACTIVE, SUSPENDED, DENIED, or FAILED	Active implies the publisher accepted the subscription. Inactive implies the publisher either expunged the subscription because of being timed out or the subscriber expunged the subscription. Suspended implies the subscription has been suspended. Denied implies the subscription can be performed (e.g. request was syntactically correct) but denied because of other reasons, e.g. user does not have a need to know. Failed implies the publisher could not process the request as made.
ERROR CODE	Text	This response is optional and may occur, providing reasons of why the subscription failed and/or denied.

Table 2. Acknowledgement Description

```
<JBI_SUBACQ>
<RES_ID>JBI-010</RES_ID>
<STREAM_ID>s001</STREAM_ID>
<STATUS>ACTIVE</STATUS>
<ERROR_CODE>NONE</ERROR_CODE>
</JBI_SUBACQ>
```

Figure 30 Example Acknowledgement

Example subscriptions and responses are shown below.

Subscription Activation XML with an X-Path Query


```
<?xml version="1.0"?>
<JBI_SUBREQ>
  <EM_ADDR>mmanning@caprarotechnologies.com</EM_ADDR>
  <USER_ID>mmanning</USER_ID>
  <RES_ID>JBI-010</RES_ID>
  <STREAM_ID>m500</STREAM_ID>
  <ACTION>ACTIVATE</ACTION>
  <Recurring value="TRUE" >
    <TimePeriod value="TRUE">
      <StartDate value="11/10/2000 16:00" />
      <EndDate   value="12/30/2000 08:00" />
    </TimePeriod>
    <RefreshRate>
      <TimeInSeconds value="0" />
    </RefreshRate>
  </Recurring>
  <Subset value="TRUE" method="XPATH" >
  <SubQuery>//Flight[FlightNo>='MC0401']</SubQuery><SubQueryFile/>
  </Subset>
</JBI_SUBREQ>
```


Corresponding Subscription Acknowledgement


```
<?xml version="1.0"?>
<JBI_SUBACQ>
        <RES_ID>JBI-010</RES_ID>
        <STREAM_ID>m500</STREAM_ID>
        <STATUS>ACTIVE</STATUS>
</JBI_SUBACQ>
```


Subscription Activation XML without an X-Path Query


```
<?xml version="1.0"?>
<JBI_SUBREQ>
  <EM_ADDR>mmanning@caprarotechnologies.com</EM_ADDR>
  <USER_ID>mmanning</USER_ID>
  <RES_ID>JBI-010</RES_ID>
  <STREAM_ID>m500</STREAM_ID>
  <ACTION>ACTIVATE</ACTION>
```

```
</JBI_SUBREQ>
```


Corresponding Subscription Acknowledgement


```
<?xml version="1.0"?>
<JBI_SUBACQ>
        <RES_ID>JBI-010</RES_ID>
        <STREAM_ID>m500</STREAM_ID>
        <STATUS>ACTIVE</STATUS>
</JBI_SUBACQ>
```


Subscription Suspension XML


```
<?xml version="1.0"?>
<JBI_SUBREQ>
  <EM_ADDR>mmanning@caprarotechnologies.com</EM_ADDR>
  <USER_ID>mmanning</USER_ID>
  <RES_ID>JBI-010</RES_ID>
  <STREAM_ID>m500</STREAM_ID>
  <ACTION>SUSPEND</ACTION>
</JBI_SUBREQ>
```


Corresponding Subscription Acknowledgement


```
<?xml version="1.0"?>
<JBI_SUBACQ>
        <RES_ID>JBI-010</RES_ID>
        <STREAM_ID>m500</STREAM_ID>
        <STATUS>SUSPENDED</STATUS>
</JBI_SUBACQ>
```


Subscription Resume XML


```
<?xml version="1.0"?>
<JBI_SUBREQ>
  <EM_ADDR>mmanning@caprarotechnologies.com</EM_ADDR>
  <USER_ID>mmanning</USER_ID>
  <RES_ID>JBI-010</RES_ID>
  <STREAM_ID>m500</STREAM_ID>
  <ACTION>RESUME</ACTION>
</JBI_SUBREQ>
```


Corresponding Subscription Acknowledgement


```
<?xml version="1.0"?>
```

```
<JBI_SUBACQ>
        <RES_ID>JBI-010</RES_ID>
        <STREAM_ID>m500</STREAM_ID>
        <STATUS>ACTIVE</STATUS>
</JBI_SUBACQ>
```

Subscription Expunge XML

```
<?xml version="1.0"?>
<JBI_SUBREQ>
  <EM_ADDR>mmanning@caprarotechnologies.com</EM_ADDR>
  <USER_ID>mmanning</USER_ID>
  <RES_ID>JBI-010</RES_ID>
  <STREAM_ID>m500</STREAM_ID>
  <ACTION>EXPUNGE</ACTION>
 </JBI_SUBREQ>
```

Corresponding Subscription Acknowledgement.

```
<?xml version="1.0"?>
<JBI_SUBACQ>
        <RES_ID>JBI-010</RES_ID>
        <STREAM_ID>m500</STREAM_ID>
        <STATUS>INACTIVE</STATUS>
</JBI_SUBACQ>
```

If there are incorrect Subscription Requests or Permission is denied then the following acknowledgements are sent.

The Subscription Request Failed and this is the acknowledgement sent.

```
<?xml version="1.0"?>
<JBI_SUBACQ>
        <RES_ID>JBI-010</RES_ID>
        <STREAM_ID>m500</STREAM_ID>
        <STATUS>FAILED</STATUS>
</JBI_SUBACQ>
```

The Subscription Request Denied because of security permissions .

```
<?xml version="1.0"?>
<JBI_SUBACQ>
        <RES_ID>JBI-010</RES_ID>
        <STREAM_ID>m500</STREAM_ID>
        <STATUS>DENIED</STATUS>
</JBI_SUBACQ>
```

The implementation of the P/S paradigm described above is implemented on two Internet servers owned by Capraro Technologies, Inc. currently located at two different locations. These servers are emulating two JBI nodes with the Internet representing a JBI network as shown in figure 15. Figure 31 depicts our current architecture. One node represents the publisher (AMC Node) and the other node represents that node that is servicing the client that is subscribing to the publication (C2 Node). Both machines contain the identical software since JBI nodes should be able to publish and subscribe to data. In our previous effort both the P/S were performed on the same server. For this effort the Internet separates the publish and subscribe processes.

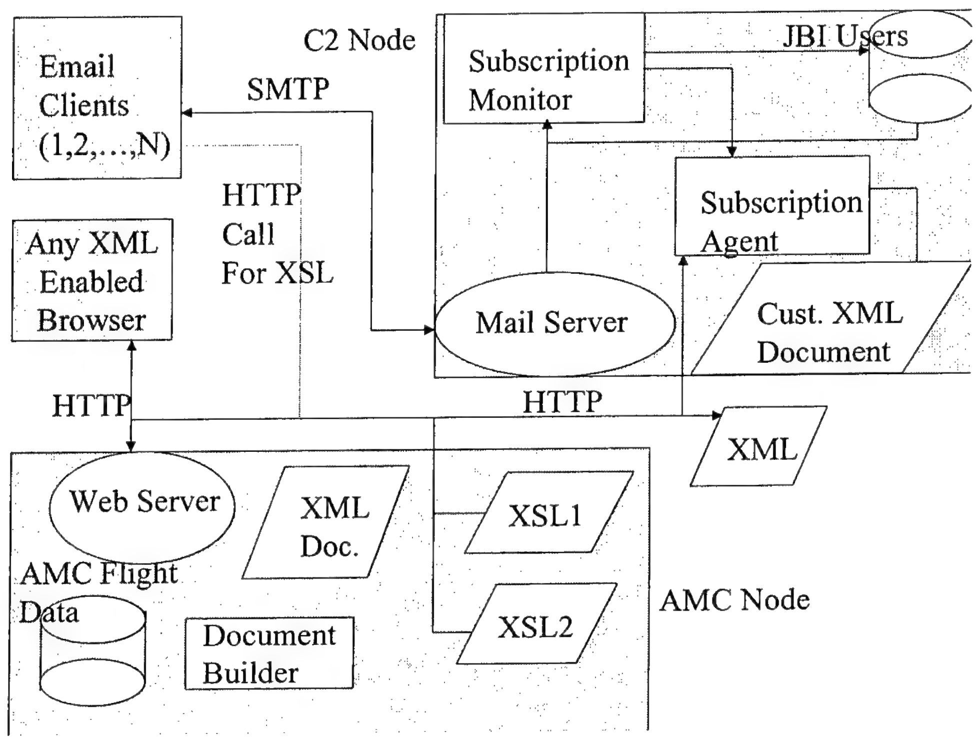


Figure 31 A 2 Nodal/Client JBI Architecture

The user can access the publication in two ways. One way is through a HTTP pull from the web site of the XML/XSL document shown in figure 28. The other way is to send email messages to our server and receive a personalized XML document. The document can then be displayed either with or without referenced style sheets (XSL) within the document. Style sheets can be stored anywhere and referenced within an XML document. For Major Marmelstein's application, a style sheet was omitted. For different applications and computing devices different style sheets can be employed. We have created two different style sheets and have created a special interface for a palm operating system environment using Java.

For the HTTP pull a user with a XML-enabled browser would only have to make a HTTP call to the publisher node (hosted on www.caprarotechnologies.com) with the correct address and a page similar to figure 28 would provide the current status of the flights. First, however, the client would receive the XML document and if a XSL reference

document is contained within the XML a second HTTP call would be made for the proper XSL for displaying the XML data. If no style sheet is present then the resultant XML would look like the data shown in figure 32.

```
<?xml version="1.0"?>
<?xml:stylesheet type="text/xsl" href="jbi010.xsl"?>
<JBI_WRAPPER>
<HEADER>
   <DOC_ID>"JBI-010"</DOC_ID>
</HEADER>
   <BODY>
      <Flight>
         <FlightNo>"MC0501"</FlightNo>
            <DepartureData>
               <DepartureLocation>"LICM (Calopezzati)"</DepartureLocation>
               <DepartureTime>"1000"</DepartureTime>
            </DepartureData>
            <DestinationData>
               <DestinationLocation>"K19G (Buffalo Coast Guard Station) "</DestinationLocation>
            </DestinationData>
            <PositionCoordinates>
               <Latitude>"44 Deg 25 Min N Latitude"</Latitude>
               <Longitude>"75 Deg 36 Min W Longitude"</Longitude>
               <PercentComplete>"95.83%"</PercentComplete>
               <LastReportTime>"1630"</LastReportTime>
            </PositionCoordinates>
           <OnTime></OnTime>
           <InTime></InTime>
      </Flight>
   </BODY>
</JBI_WRAPPER>
```

Figure 32 Example XML Results.

If the user sends an email to a subscription process, then the Mail Server grabs the mail, validates the user, analyzes the subscription and responds with a subscription status. It also energizes the subscription agent to acquire the proper publication. In our situation the publication is contained on another JBI node. It uses an HTTP call to acquire the proper XML document. Depending upon the user and their subscription it customizes the XML document. It then formulates an email response and sends the XML document as an email attachment. As long as the subscription is not suspended or expunged the subscription agent will monitor the publisher. If the data the subscription agent has been monitoring has changed then it will email the updated document to the subscriber. This continues until the subscriber expunges the subscription or it expires. To display the XML data within an XML-enabled browser, the browser can retrieve a style sheet from the URL referenced within the XML document. Currently we have different XML documents with different XSL references dependent upon the user and their computing device.

For the mobility of the future military personnel we developed a Java interface for a Palm and Windows CE operating system (OS). The current browser on the CE system is XML enabled and can perform a HTTP call to receive information similar to a PC. The Palm's browser is not XML enabled as yet, if an HTTP call is made then it receives the XML as text only and the XSL is ignored. However, the Palm and CE devices can send and receive email using additional commercial off the shelf software. We developed an interface in Java that produces subscription management documents in XML format. These documents are emailed using the email client software. Because the Palm is not XML enabled the software on the server sends a text version of the proper response. Figures 33-35 provide a view of the Palm JBI subscription email interface.

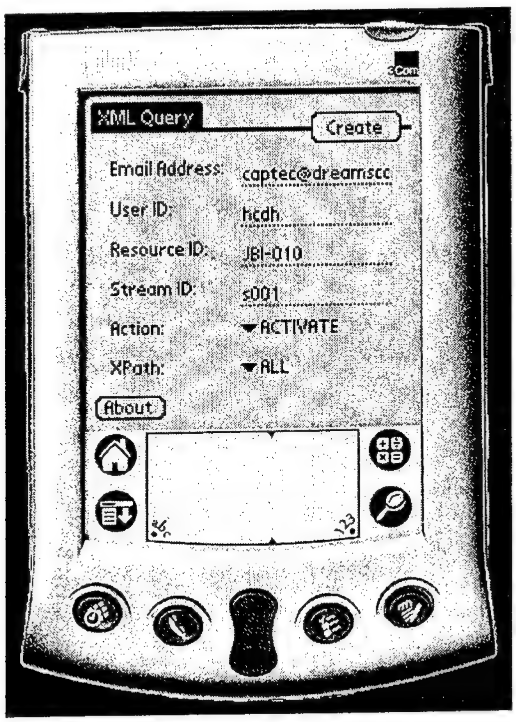


Figure 33 Palm email Generation Screen

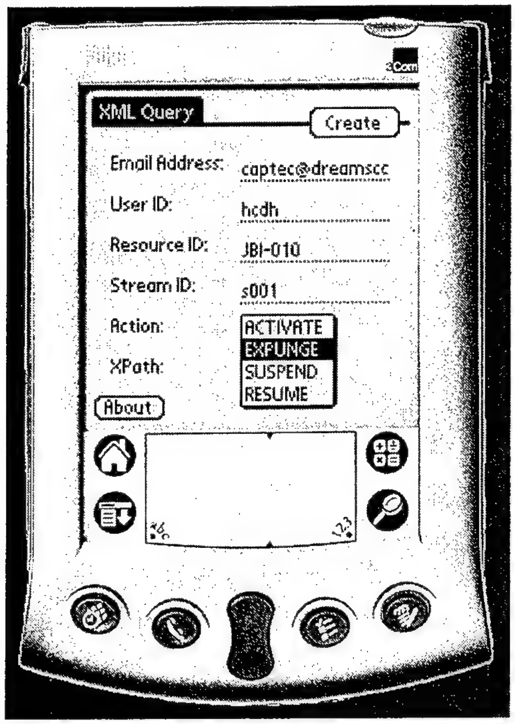


Figure 34 Palm email Action Screen

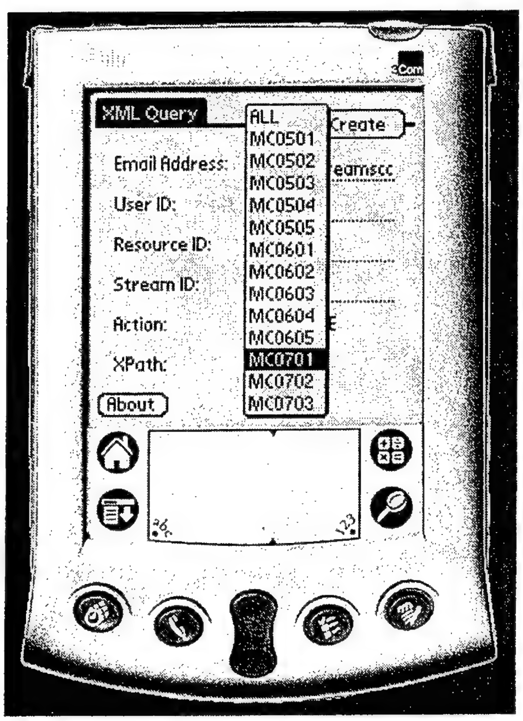


Figure 35 Palm email Choosing Flight Number for Xpath Query

Section 6 Summary and Future Work

This final report has provided an overview of the work we have performed and a description of the demonstrations we developed. A challenge problem was defined for the Air Mobility Command (AMC) for the 2010 time frame. The ability of having 3-D images of airports provided to a pilot either while in flight or before they leave to fly into an area or airstrip is provided. We developed a proof of concept demonstration using Eglin AFB as an example. The demonstration uses US Geological Society (USGS) databases for the elevation data, virtual reality modeling language (VRML) as the development tool, and the warping of images to the terrain model for scene creation. We also showed the capability of modifying images for additional reality, such as buildings, or data that was not captured in an image such as damage from a recent bombing. A software architecture was developed and described and a description of our VRML demonstration provided.

During the course of the effort we were asked to evaluate the Joint Battlespace Infosphere (JBI) model and determine its role in the goals of this effort. A brief overview of the JBI was presented and it was determined that the JBI publish and subscribe (P/S) paradigm was a good fit in providing a solution to the AMC defined problem domain. A JBI architecture is described which allows USAF personnel to acquire AMC flight information and request and create VRML scenes of locations anywhere in the world. A hypothetical scenario is provided as to how the process would function in a combat situation.

During a previous effort it was demonstrated how USAF personnel could access large data and knowledge bases with hand-held computing devices (HCD). During that effort the AMC problem domain was used to demonstrate this capability related to aircraft flight data. In concert with this effort, Maj. Marmelstein, of the IF Directorate was working on an email JBI paradigm emphasizing the use of subscriptions and fuselets. Collaboration between these efforts was described in building a JBI email paradigm. A demonstration leveraging the previous work of building flight scenario simulations of AMC flight status data was used. A description of the results of this previous effort is provided, followed by a description of the integration of our two efforts.

The JBI email paradigm example uses W3C technologies. A published XML document and style sheet is described along with the subscription and acknowledgement descriptions that were developed. Major Marmelstein's software generates the subscription requests and emails it to one of two nodes that were developed within this effort. A second node performs the simulation and generates the publication. The system is described and numerous XML subscriptions and acknowledgement responses are provided.

To emphasize the power of this approach we developed a Java application for the Palm HCD allowing one to generate XML subscription management documents which are then delivered to the appropriate node. The information can also be gathered using a Windows CE device or any device with a browser or email capability. It should also be

noted that the publication capability was developed on a Linix machine running Java, the HCD subscription software is running on the Palm operating system, and Maj. Marmelstein's software was developed with Microsoft products and tools. This exercise showed that we could define our interfaces in the open system W3C standards and the system did work.

The current software system is for demonstration purposes only. It is fragile and only allows one user at a time to participate in managing subscriptions. A more robust version should be built allowing multiple users, access with implementing security, priority, computing device dependencies, encryption and publication controls enhancements. Management of subscription issues should be studied along with user profile requests especially when considering other domains such as voice, video, and images. The architecture shown in figure 31 should be enhanced and made more robust by leveraging the results of current USAF, W3C and DARPA initiatives.

In parallel with these suggested enhancements we should use the AMC domain to demonstrate the added capability and value obtained. We should also seek out other domains that can reap the benefits of the JBI architecture. Some suggestions are the integration of multiple sensors, retrieving data relevant to planning, tracking people, equipment, bombs, etc. Building demonstrations in other domains using these tools will test the robustness of our design and indicate where the system software and the application domain should be separated.

Acknowledgements

The authors would like to recognize the efforts of numerous people. We would like to thank Mr. John Spina, Dr. Raymond Liuzzi, Major Robert Marmelstein, and their management for providing the resources and guidance in the pursuit of our goals. We would also like to recognize the guidance we received from Dr. John Evanowski, Ms. Pat Baskinger, and Ms. Mary Carol Chruscicki for their help in understanding AMC's problem domain. Last on our list we owe a debt of gratitude to Dr. P. Bruce Berra who worked with us on the JBI summer study program and the employees of Capraro Technologies, Inc. that provided their talents and creativity in building the prototype software described in this document. These people are Mr. Christopher Capraro, Ms. Michelle Manning, Mr. Anthony Macera, Ms, Sarah Schiavone, Mr. Mark D'Agostino, Mr. Brice DeWire, and Mr. Ashish Patel. Thank you all.

References

1.Capraro, G.T., "A Data Management System Problem Specification Model," RADC—TR—73-193, June 1973.

2. Capraro, G.T., Berra, P. B., "A Data Base Management Problem Specification Model, "AFIPS Conference Proceedings," Vol. 43, 1974, pp. 53—56.

3. Capraro, G.T., "A Data Base Management Modeling Technique and Special Function Hardware Architecture," Unpublished Doctoral Dissertation, Syracuse University, February 1978.

4. Capraro, G.T., Berra, P.B., "A Data Base Management Modeling Technique and Special Function Hardware Architecture," RADC—TR—79—14, January 1979.

5. Lazzara, L. V., Marcinkowski, J. M., Capraro, G.T., and White, R. C., "Knowledge Based and Database System Integration: A Base Line for a Design Methodology," RADC-TR-86-132, September 1986.

6. Lazzara, L.V., Marcinkowski, J., Capraro, G.T., Liuzzi, R., and White, R., "A Framework for Extending Information Systems With Knowledge Based Processing Capabilities," Proceedings of the Computer Software and Applications Conference, October 1986.

References 7&8 removed per Project Engineer request.

9. Capraro, G.T., Cheung, C., and Liuzzi, R. A., "Large Knowledge Based Systems: An Efficient Approach," Proceedings of the Annual AI Systems in Government Conference, March 1989.

10. Capraro, G.T., Siarkiewicz, K., "Computational Electromagnetics' Future Database Architecture," 13th Annual Review of Progress in Applied Computational Electromagnetics, March 1997.

11. Capraro, G.T., "Integrated Computational Environment (ICE)," RL-TR-97-70, August 1997.

12. Kwasowsky, B., Capraro, G. T., Berdan, G. B., Capraro, C. T., "Remote Data Entry and Retrieval for Law Enforcement," First Annual Symposium on Enabling Technologies for Law Enforcement and Security, November 1996.

13. Capraro, G. T., "Hand-Held Computing Devices and Large Knowledge Bases," AFRL-IF-RS-TR-1998-205, November 1998.

14. Capraro, G., and Berdan, G., "Intelligent Access To Large Knowledge Bases From Heterogeneous Devices Using Multiple Protocols", AFRL-IF-RS-TR.

15. United States Air Force Scientific Advisory Board, "Report on Information Management to Support the Warrior," SAB-TR-98-02, December 1998

16.United States Air Force Scientific Advisory Board, "Report on Building the Joint Battlespace Infosphere," SAB-TR-99-02, November 1999

17, Airfield Initiative Program, http://164.214.2.59/airfield/english-pamphlet.rtf

☆U.S. GOVERNMENT PRINTING OFFICE: 2001-710-038-10188

MISSION
OF
AFRL/INFORMATION DIRECTORATE (IF)

The advancement and application of Information Systems Science and Technology to meet Air Force unique requirements for Information Dominance and its transition to aerospace systems to meet Air Force needs.